

# UNIVERSIDAD TÉCNICA DEL NORTE

Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Mantenimiento Eléctrico



## IMPLEMENTACIÓN DE UN GUANTE HÁPTICO CON RETROALIMENTACIÓN DE FUERZA

Trabajo de grado previo a la obtención del título de Ingeniero en Mantenimiento  
Eléctrico

Autor:

Joseph Sebastián Albuja Pozo

Director:

Ing. Francisco Roberto Naranjo Cobo MSc.

Ibarra – Ecuador

2023



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN

#### A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

| DATOS DE CONTACTO    |   |                 |            |
|----------------------|---|-----------------|------------|
| CÉDULA DE IDENTIDAD: | 100417484-1                                 |                 |            |
| APELLIDOS Y NOMBRES: | ALBUJA POZO JOSEPH SEBASTIAN                |                 |            |
| DIRECCIÓN:           | Sánchez y Cifuentes 8-38 y Juan José Flores |                 |            |
| EMAIL:               | jsalbuja@utn.edu.ec                         |                 |            |
| TELÉFONO FIJO:       |   | TELÉFONO MÓVIL: | 0982209646 |

| DATOS DE LA OBRA            |  |
|-----------------------------|--|
| TÍTULO:                     | Implementación de un guante háptico con retroalimentación de fuerza            |
| AUTOR (ES):                 | ALBUJA POZO JOSEPH SEBASTIAN   |
| FECHA: DD/MM/AAAA           | 24/05/2023   |
| SOLO PARA TRABAJOS DE GRADO |  |
| PROGRAMA:                   | <input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO |
| TÍTULO POR EL QUE OPTA:     | Ingeniería en Mantenimiento Eléctrico  |
| ASESOR /DIRECTOR:           | Ing. Francisco Roberto Naranjo Cobo MSc.                                       |

#### 2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el

contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 30 días del mes de Mayo de 2023

**EL AUTOR:**

(Firma) 

Nombre: Sebastián Albija.



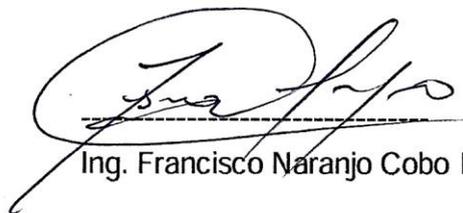
**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS**  
**APLICADAS**

**ACEPTACIÓN DEL DIRECTOR**

Ing. Francisco Naranjo Cobo MSc.

CERTIFICA

Que después de haber examinado el presente trabajo de investigación elaborado por el señor estudiante: Albuja Pozo Joseph Sebastián, certifico que ha cumplido con las normas establecidas en la elaboración del trabajo de investigación titulado: **“IMPLEMENTACIÓN DE UN GUANTE HÁPTICO CON RETROALIMENTACIÓN DE FUERZA”** para la obtención del título de Ingeniero en Mantenimiento Eléctrico.



Ing. Francisco Naranjo Cobo MSc.

DIRECTOR DE TRABAJO DE GRADO

## **Dedicatoria**

Dedico este trabajo a todas las personas que han formado parte de mi camino hacia la realización de este proyecto. En primer lugar, a mi familia por su amor incondicional y apoyo constante durante todo el proceso. A mis amigos y compañeros de estudio, por sus palabras de aliento y motivación cuando más lo necesitaba. A mis profesores y asesores, por compartir su conocimiento y guiarme en el camino hacia la excelencia académica. Y finalmente, a mí mismo, por no rendirme ante los obstáculos y seguir luchando por mis sueños.

¡Gracias a todos por ser parte de este logro!

## **Agradecimiento**

Me gustaría expresar mi sincero agradecimiento a todas las personas que me brindaron su apoyo y colaboración en la realización de esta tesis. En primer lugar, a mi director de tesis, por su paciencia, sabiduría y experiencia en la materia, así como por sus valiosos consejos y orientación en todo momento. Agradezco también a los profesores y expertos que me brindaron su tiempo y conocimiento para enriquecer esta investigación.

Asimismo, quiero agradecer a mis amigos y compañeros de clase, quienes me brindaron su ánimo y aliento cuando más lo necesitaba. Agradezco especialmente a mi familia, por su amor y apoyo incondicional, y por brindarme las condiciones necesarias para poder completar esta etapa de mi formación profesional.

Finalmente, deseo agradecer a todas aquellas personas que, de una u otra manera, contribuyeron a la realización de este trabajo. Espero que sea de utilidad y aporte al avance del conocimiento en el campo de la electricidad y la tecnología.

## Tabla de Contenidos

|  |                                      |
|--|--------------------------------------|
| <b>ACEPTACIÓN DEL DIRECTOR .....</b>   | <b>¡Error! Marcador no definido.</b> |
| Dedicatoria.....   | V                                    |
| Agradecimiento .....   | VI                                   |
| Tabla de Contenidos .....  | VII                                  |
| Índice de Figuras .....  | X                                    |
| Índice de tablas .....   | XI                                   |
| Resumen.....   | XII                                  |
| Abstract.....  | XIII                                 |
| Introducción.....  | 14                                   |
| Contextualización.....   | 14                                   |
| Planteamiento del problema.....  | 15                                   |
| Justificación del trabajo.....   | 16                                   |
| Alcance del trabajo.....   | 16                                   |
| Objetivos Específicos .....  | 17                                   |
| <b>CAPÍTULO 1 .....</b>  | <b>18</b>                            |
| <b>Tecnologías relacionadas con la implementación de un guante háptico con retroalimentación de fuerza .....</b> | <b>18</b>                            |
| <b>1.1 La Háptica .....</b>  | <b>18</b>                            |
| 1.1.1 Interfaces Hápticas .....  | 19                                   |
| 1.1.2 Sistemas Hápticos .....  | 20                                   |
| 1.1.3 Biomecánica de la mano.....  | 21                                   |
| <b>1.3 Guante de tipo Exoesqueleto.....</b>  | <b>24</b>                            |
| <b>1.4 Sistemas de Control Háptico .....</b>   | <b>25</b>                            |
| <b>1.5 Hardware de Control.....</b>  | <b>25</b>                            |
| 1.5.1 Tarjeta de desarrollo.....   | 25                                   |
| 1.5.1.1 Arduino Mega.....  | 26                                   |
| 1.5.2 Sensor Infrarrojo de distancia marca Sharp .....   | 26                                   |
| 1.5.3 Servomotor .....   | 27                                   |

|                   |  |           |
|-------------------|--|-----------|
| 1.5.4             | PWM Vibrador Arduino .....   | 29        |
| 1.5.5             | Sensor Flexométrico .....  | 29        |
| 1.5.6             | Motorreductores DC.....  | 30        |
| 1.5.7             | Driver TB6612fNG .....   | 31        |
| 1.5.8             | Batería Li-Po.....   | 31        |
| 1.5.9             | Módulo de carga BMS 3S 10A.....  | 32        |
| <b>1.6</b>        | <b>Módulos de comunicación</b> .....                                   | <b>32</b> |
| 1.6.1             | Módulo LoRa .....  | 32        |
| 1.6.2             | Dragino Shield LoRa 433 MHz.....                                       | 33        |
| 1.6.3             | Giroscopio .....   | 34        |
| <b>1.7</b>        | <b>Servicio de impresión 3D</b> .....                                  | <b>35</b> |
| <b>1.8</b>        | <b>Software de Desarrollo</b> .....                                    | <b>36</b> |
| 1.8.1             | Arduino IDE .....  | 36        |
| <b>CAPÍTULO 2</b> | .....  | <b>38</b> |
|                   | <b>Diseño del guante háptico con retroalimentación de fuerza</b> ..... | <b>38</b> |
| <b>2.1</b>        | <b>Metodología</b> .....   | <b>38</b> |
| <b>2.2</b>        | <b>Características de guante háptico</b> .....                         | <b>38</b> |
| <b>2.3</b>        | <b>Tipo de guante háptico</b> .....                                    | <b>39</b> |
| <b>2.4</b>        | <b>Tipo de vehículo</b> .....  | <b>42</b> |
| <b>2.5</b>        | <b>Componentes que integran el guante háptico</b> .....                | <b>43</b> |
| 2.5.1             | Controlador para el guante háptico .....                               | 43        |
| 2.5.2             | Sensores para el guante háptico.....                                   | 44        |
| 2.5.3             | Comunicación para el guante háptico .....                              | 44        |
| 2.5.4             | Actuadores o motores para el guante háptico .....                      | 45        |
| 2.5.5             | Giroscopio .....   | 46        |
| 2.5.6             | Capacidad de la batería para el guante háptico .....                   | 46        |
| <b>2.6</b>        | <b>Componentes que integran el vehículo</b> .....                      | <b>47</b> |
| 2.6.1             | Controlador para el vehículo .....                                     | 47        |
| 2.6.2             | Motores o actuadores .....   | 48        |

|                        |   |           |
|------------------------|---|-----------|
| 2.6.3                  | Sensores para el vehículo.....  | 49        |
| 2.6.4                  | Capacidad de la batería para el vehículo .....                                | 50        |
| <b>2.7</b>             | <b>Diseño electrónico del guante háptico.....</b>                             | <b>51</b> |
| <b>2.8</b>             | <b>Diseño electrónico del vehículo .....</b>                                  | <b>53</b> |
| <b>2.9</b>             | <b>Diagrama de bloques.....</b>   | <b>56</b> |
| <b>2.10</b>            | <b>Software .....</b>   | <b>57</b> |
| <b>2.11</b>            | <b>Flujograma.....</b>  | <b>57</b> |
| <b>CAPÍTULO 3</b>      | <b>.....</b>  | <b>59</b> |
|                        | <b>Implementación del guante háptico con retroalimentación de fuerza.....</b> | <b>59</b> |
| <b>3.1</b>             | <b>Impresión 3D del guante háptico con retroalimentación de fuerza.....</b>   | <b>59</b> |
| <b>3.2</b>             | <b>Ensamblado del exoesqueleto.....</b>                                       | <b>61</b> |
| <b>3.3</b>             | <b>Distribución y montaje de los componentes electrónicos .....</b>           | <b>64</b> |
| 3.3.1                  | Montaje de los componentes del guante háptico en el exoesqueleto.....         | 64        |
| 3.3.2                  | Montaje de los componentes del vehículo .....                                 | 68        |
| <b>3.4</b>             | <b>Desarrollo de Software .....</b>   | <b>71</b> |
| <b>3.5</b>             | <b>Trama de datos .....</b>   | <b>71</b> |
| <b>3.6</b>             | <b>Pruebas .....</b>  | <b>73</b> |
| <b>3.7</b>             | <b>Resultados .....</b>   | <b>80</b> |
| 3.7.1                  | Desempeño del controlador .....   | 80        |
| 3.7.2                  | Desempeño de los actuadores.....  | 81        |
| 3.7.3                  | Desempeño de la comunicación.....   | 82        |
| 3.7.4                  | Desempeño del giroscopio.....   | 83        |
| 3.7.5                  | Desempeño de los sensores.....  | 84        |
| 3.7.6                  | Duración de la batería .....  | 85        |
| <b>CONCLUSIONES</b>    | <b>.....</b>  | <b>87</b> |
| <b>RECOMENDACIONES</b> | <b>.....</b>  | <b>88</b> |
| Bibliografía           | .....   | 89        |

## Índice de Figuras

|                  |    |
|------------------|----|
| Figura. 1 .....  | 18 |
| Figura. 2 .....  | 20 |
| Figura. 3 .....  | 21 |
| Figura. 4 .....  | 22 |
| Figura. 5 .....  | 23 |
| Figura. 6 .....  | 23 |
| Figura. 7 .....  | 24 |
| Figura. 8 .....  | 25 |
| Figura. 9 .....  | 26 |
| Figura. 10 ..... | 27 |
| Figura. 11 ..... | 28 |
| Figura. 12 ..... | 28 |
| Figura. 13 ..... | 29 |
| Figura. 14 ..... | 30 |
| Figura. 15 ..... | 30 |
| Figura. 16 ..... | 31 |
| Figura. 17 ..... | 32 |
| Figura. 18 ..... | 33 |
| Figura. 19 ..... | 34 |
| Figura. 20 ..... | 34 |
| Figura. 21 ..... | 35 |
| Figura. 22 ..... | 36 |
| Figura. 23 ..... | 38 |
| Figura. 24 ..... | 40 |
| Figura. 25 ..... | 40 |
| Figura. 26 ..... | 41 |
| Figura. 27 ..... | 41 |
| Figura. 28 ..... | 42 |
| Figura. 29 ..... | 52 |
| Figura. 30 ..... | 54 |
| Figura. 31 ..... | 56 |
| Figura. 32 ..... | 58 |
| Figura. 33 ..... | 60 |
| Figura. 34 ..... | 60 |
| Figura. 35 ..... | 61 |
| Figura. 36 ..... | 62 |

|                         |    |
|-------------------------|----|
| <b>Figura. 37</b> ..... | 62 |
| <b>Figura. 38</b> ..... | 63 |
| <b>Figura. 39</b> ..... | 63 |
| <b>Figura. 40</b> ..... | 64 |
| <b>Figura. 41</b> ..... | 65 |
| <b>Figura. 42</b> ..... | 65 |
| <b>Figura. 43</b> ..... | 66 |
| <b>Figura. 44</b> ..... | 67 |
| <b>Figura. 45</b> ..... | 67 |
| <b>Figura. 46</b> ..... | 68 |
| <b>Figura. 47</b> ..... | 69 |
| <b>Figura. 48</b> ..... | 70 |
| <b>Figura. 49</b> ..... | 70 |
| <b>Figura. 50</b> ..... | 73 |
| <b>Figura. 51</b> ..... | 74 |
| <b>Figura. 52</b> ..... | 75 |
| <b>Figura. 53</b> ..... | 76 |
| <b>Figura. 54</b> ..... | 76 |
| <b>Figura. 55</b> ..... | 77 |
| <b>Figura. 56</b> ..... | 78 |
| <b>Figura. 57</b> ..... | 78 |
| <b>Figura. 58</b> ..... | 79 |
| <b>Figura. 59</b> ..... | 80 |
| <b>Figura. 60</b> ..... | 82 |
| <b>Figura. 61</b> ..... | 83 |
| <b>Figura. 62</b> ..... | 84 |
| <b>Figura. 63</b> ..... | 85 |

## Índice de tablas

|               |    |
|---------------|----|
| Tabla 1.....  | 19 |
| Tabla 2.....  | 27 |
| Tabla 3.....  | 42 |
| Tabla 4.....  | 43 |
| Tabla 5.....  | 44 |
| Tabla 6.....  | 44 |
| Tabla 7.....  | 45 |
| Tabla 8.....  | 46 |
| Tabla 9.....  | 46 |
| Tabla 10..... | 48 |
| Tabla 11..... | 48 |
| Tabla 12..... | 49 |
| Tabla 13..... | 50 |
| Tabla 14..... | 71 |
| Tabla 15..... | 75 |
| Tabla 16..... | 81 |
| Tabla 17..... | 81 |
| Tabla 18..... | 86 |

## Resumen

El desarrollo de la tecnología con las personas en el mundo se basa mayormente a los sentidos del oído y la vista, las interfaces hápticas aprovechan el sentido del tacto, permitiendo al usuario percibir y manipular objetos con una retroalimentación de la información obtenida remotamente, así resolviendo los problemas de interrelación hombre máquina HMI y mejorando la comunicación de los humanos con las nuevas tecnologías.

El objetivo de esta investigación es la implementación de un guante háptico con retroalimentación de fuerza utilizando la metodología en espiral la cual permite seguir un proceso ordenado e interactuar con los mejores procesos para obtener los mejores resultados, así realizando pruebas y ajuste de diseños existentes en donde se obtiene que los guantes hápticos comerciales se pueden comunicar en un entorno virtual, pero también pueden ser implementados con tecnologías de acción y reacción con las personas como el manejo de vehículos electrónicos, que tiene un costo menor a los guantes de comunicación con entornos virtuales, pero cumple con los requerimientos de la háptica en la actualidad.

El guante háptico con retroalimentación de fuerza maneja un pequeño vehículo no tripulado que por medio de los sensores permite al usuario percibir y sentir mediante acciones y vibraciones los obstáculos que se presentan en el campo de manejo, todo esto dirigido a la mano derecha de la persona.

**Palabras clave:** interfaces hápticas, HMI, remotamente, retroalimentación.

## **Abstract**

The development of technology with people in the world is based mostly on the senses of hearing and sight, haptic interfaces take advantage of the sense of touch, allowing the user to perceive and manipulate objects with feedback of information obtained remotely, thus solving HMI man-machine interrelation problems and improving human communication with new technologies.

The objective of this research is the implementation of a haptic glove with force feedback using the spiral methodology which allows following an orderly process and interacting with the best processes to obtain the best results, thus testing and adjusting existing designs where it is obtained that commercial haptic gloves can communicate in a virtual environment but can also be implemented with action and reaction technologies with people such as electronic vehicle management, which has a lower cost than communication gloves with virtual environments but complies with today's haptic requirements.

The haptic glove with force feedback manages a small, unmanned vehicle that, through sensors, allows the user to perceive and feel through actions and vibrations the obstacles that appear in the driving field, all directed to the right hand of the person.

**Keywords:** haptic interfaces, HMI, remotely, feedback.

## Introducción

### Contextualización

En la actualidad desde la aparición de las nuevas tecnologías, el entorno social ha cambiado drásticamente debido a la interacción del hombre y su entorno, teniendo así un nuevo inicio en la era de la comunicación con la ciencia. Las personas cada vez buscan interactuar más con las máquinas, software y computadoras de última generación, permitiendo a la humanidad tener una comunicación más eficiente con interfaces y dispositivos tecnológicos donde el usuario puede compartir datos en tiempo real permitiendo tener el control total de artefactos o sistemas computacionales (Copadata, 2021).

La tecnología háptica se ha centrado en el diseño e implementación de prototipos de uso en entornos virtuales como: entretenimiento, multimedia, ingeniería, robótica y simuladores, involucrando al tacto como uno de los sentidos más importantes para el ser humano sin la necesidad de depender de la visión (Altamirano-Droguett, 2019).

La interacción de las personas mediante interfaces hápticas cumple un papel importante debido a que además de enviar datos al computador se podrá recibir información proveniente del ambiente virtual, todo esto mediante una sensación física transmitida por fuerzas de torsión a las articulaciones de la mano u otras partes del cuerpo, denominándose así guantes con retroalimentación de fuerza (Quito Bacuilima, 2022).

A nivel internacional se ha estudiado el uso y la transferencia de mecanismos para la acción de fuerzas entre ellos está la actuación neumática y a través de cables, cada uno de ellos cumpliendo un objetivo específico como el tamaño del dispositivo, aumento o disminución de peso o una retroalimentación con mayor fuerza de torsión (Arri, 2019).

Las interfaces virtuales y hápticas también han aportado a la medicina en varios aspectos, uno de estos aportes es la facilidad del cateterismo; es decir mediante un sistema de asistencia de control de fuerza y precisión del catéter en prácticas médicas y de complejidad evitan que se dañen las venas del paciente durante el proceso, todo esto aplicado en un dispositivo llamado "HapCatch" (Sánchez J. P., 2019).

Por otro lado, en Puebla México un grupo de personas implementaron la realidad virtual junto con un sistema háptico para proyección de películas virtuales en el cine para personas no videntes (Medellín, 2011). Ayudando a la comunidad de personas privadas de la visión a que puedan tener una vida activa normal y volviéndolos parte de la comunidad.

Zhou ha diseñado un exoesqueleto con retroalimentación háptica denominado “RLM GLOVE” el cual básicamente se trata de un guante con sistema mecatrónico y retroalimentación con micromotores, encargados de transmitir la fuerza de retroalimentación hacia los dedos de la mano y permitiendo su libre movimiento. Dicho dispositivo se logró alcanzar mediante pruebas de navegación con un robot móvil e interfaces hápticas aplicadas a cada dedo de la mano, con la finalidad de mejorar la coordinación del robot y la teleoperación de la persona con el dispositivo (du Plessis, 2021).

En la Universidad de las Fuerzas Armadas ESPE en el Ecuador se han desarrollado varios proyectos que emplean la realidad virtual para mejorar la rehabilitación de pacientes que sufren de accidentes cerebrovasculares, entre ellos está la implementación de una herramienta virtual mediante un dispositivo háptico para la rehabilitación de pacientes con daño cerebral, el cuál básicamente se basa de un dispositivo háptico comercial GEOMAGIC TOUCH que se enfoca en el diseño de una interfaz de usuario con ejercicios físicos y una base de datos con respecto a los movimientos que ha realizado durante la rehabilitación, para su posterior seguimiento y evaluación (Andres, 2017).

### **Planteamiento del problema**

Las personas usan los sentidos en la vida diaria para percibir el ambiente que lo rodea, trabajando con la tecnología virtual usualmente se cumple la retroalimentación de la vista y el oído, por ello la telepresencia de los usuarios sigue siendo baja, donde una completa telepresencia se logra alcanzar cuando la persona se siente presente en una ubicación remota o virtual. En este punto, las interfaces hápticas tienen gran importancia, ya que estas aprovechan el sentido del tacto, permitiendo al usuario percibir y manipular objetos con una retroalimentación de la información obtenida en un entorno virtual, mejorando la telepresencia del mismo (Arri, 2019).

Dicho de una mejor forma las personas que cuentan con dificultades o han sufrido accidentes en donde hayan sido afectados sus sentidos como visión, tacto y audición, tienen la dificultad en poder acoplarse nuevamente a la vida rutinaria. En donde por la ausencia de sus sentidos dejan de sentirse personas normales y tienden a aislarse, dejando de lado la interacción con el entorno que los rodea.

## **Justificación del trabajo**

En Ecuador, justamente en la provincia de Imbabura; la aplicación de interfaces hápticas no es muy común, debido a que son muy escasas y sus dispositivos virtuales únicamente se enfocan en la telepresencia con las máquinas tecnológicas a través del oído y la vista, por ello la implementación de un guante háptico para mejorar la telepresencia de las personas controlando un vehículo no tripulado mediante una retroalimentación de fuerza para evitar obstáculos a través del tacto, será de gran utilidad para las personas que hayan sufrido accidentes y tengan falencias en cuanto a sus sentidos de la vista, audición y tacto, debido a que promocionará a que nuevamente puedan integrarse a su vida rutinaria sin dificultades.

El guante háptico tiende a la posibilidad de futuras mejoras con respecto a tecnología e interacción, debido a que podrá ser sometido con otros métodos virtuales para su desarrollo en cuanto a telepresencia con dispositivos virtuales, con la finalidad de que sean de gran ayuda en diversas aplicaciones dentro de la sociedad. Las aplicaciones pueden orientarse a distintos beneficiarios permitiendo su personalización de acuerdo con las necesidades de cada usuario, entre ellos puede ser de uso personal, entretenimiento o de rescate de personas en lugares peligrosos, siendo un dispositivo seguro y libre de elementos que pueden afectar a las personas y al ambiente.

## **Alcance del trabajo**

El diseño de este proyecto háptico va a permitir a las personas interactuar de una forma novedosa con el mundo, manipulando su entorno con las manos a través de un guante ajustable a cualquier tamaño, para que todos sin importar su edad puedan hacer uso de él. Aportando al aprendizaje práctico a través de las interfaces motrices y teleoperables en la actualidad, llevando sensaciones transmitidas del dispositivo a cada dedo de nuestra mano, donde se utilizará microcontroladores por medio de sensores flexométricos y servomotores para la retroalimentación, así determinando el doblar de los dedos y la palma de la mano, también un dispositivo de vibración para el aviso de aproximación de obstáculos en la teleoperación del vehículo no tripulado, todo esto montado en un exoesqueleto impreso en 3D.

## **Objetivo General**

Implementar un guante háptico con retroalimentación de fuerza controlado a través de un exoesqueleto automatizado por hardware para la operación de un vehículo no tripulado.

## **Objetivos Específicos**

1. Describir las tecnologías relacionadas con la implementación de un guante háptico con retroalimentación de fuerza.
2. Diseñar un guante háptico con retroalimentación de fuerza.
3. Implementar el guante háptico con retroalimentación de fuerza.

# CAPÍTULO 1

## Tecnologías relacionadas con la implementación de un guante háptico con retroalimentación de fuerza

El presente capítulo contiene información conceptual y bibliográfica con respecto a conceptos técnicos de los sistemas hápticos, sistemas de control y de fuerza, sensores y softwares de desarrollo, finalmente un análisis de las tecnologías descritas en los conceptos relacionadas con la implementación de un guante háptico con retroalimentación de fuerza.

### 1.1 La Háptica

El término háptica se refiere a sentir y manipular nuestro entorno a través del tacto siendo un componente de comunicación para las distintas personas. Inicialmente, los psicólogos utilizaron el término "háptica" para estudiar esta sensación humana, con el transcurso de los años fue necesario cambiar el término debido a los avances tecnológicos relacionados con el tacto. La nueva definición de háptica se refiere a la manipulación de objetos a través del tacto por humanos, máquinas o una combinación de ambos, que puede involucrar entornos operativos virtuales o remotos (Peña-Cortés, 2022). En la Figura 1 podemos observar una interpretación de la háptica con los entornos virtuales.

**Figura. 1**  
*Tecnología háptica*



**Fuente.** Tomado de (*Cableworld, 2019*)

Según Soledad Ballesteros define al sistema háptico como "la percepción del individuo del mundo actual a su cuerpo mediante el uso de su propio cuerpo" dándonos a saber que el sistema de percepción sensorial de nuestro organismo puede tener un efecto directo con el mundo que está percibiendo cada persona, también está asociado el sistema háptico con el

concepto de contacto activo donde se refiere a que al hacer uso de un instrumento nuestra percepción háptica se expande tal como cuando usamos un bastón y nuestra percepción es transferida directamente hacia el final del mismo (Ballesteros & Martínez Niño, 2021).

Las interfaces táctiles permiten a las personas interactuar con entornos reales o virtuales utilizando habilidades mecánicas, sensoriales, motoras y cognitivas. La interacción consta de una o más operaciones, que se pueden dividir en: movimiento y percepción (Borrego González, 2020)

La percepción háptica también se puede dividir en características de percepción cinestésica y táctil de acuerdo con las características psicológicas. La percepción táctil describe la captación basada en receptores sensoriales sobre la piel humana (Repullo Menéndez, 2021).

La percepción cinestésica describe la percepción del estado del sistema de movimiento del cuerpo humano, especialmente la posición de las articulaciones, la alineación de las extremidades, la dirección del cuerpo y la tensión de los músculos (Frutos, 2019). En la Tabla 1 podemos observar los tipos de estimulación táctil con respecto a la cinestésica.

**Tabla 1**

*Percepción háptica*

| <b>TÁCTIL</b>                 | <b>CINESTÉSICA</b>                |
|-------------------------------|-----------------------------------|
| <b>Estimulación mecánica</b>  | Fuerza / torsión                  |
| <b>Estimulación térmica</b>   | Orientación del cuerpo            |
| <b>Estimulación eléctrica</b> | Alineamiento de extremidades      |
| <b>Estimulación química</b>   | Posicionamiento de articulaciones |

**Fuente.** Tomado de (Hatzfeld, 2010)

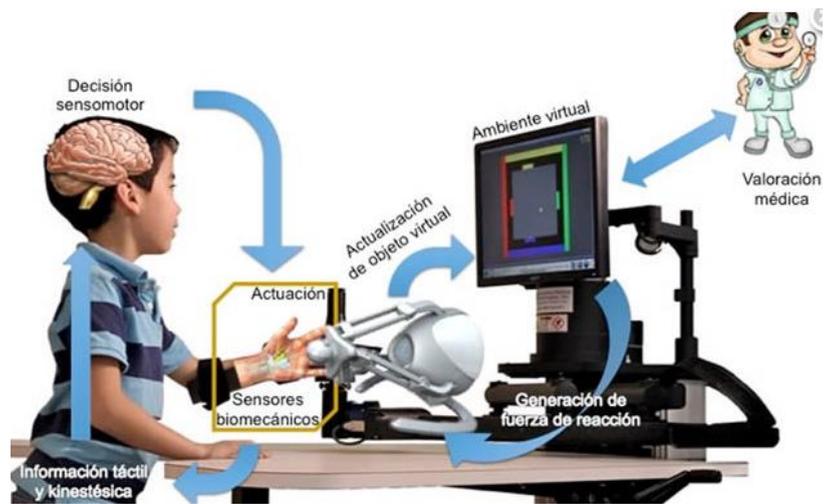
Estos dos conjuntos trabajan para proveer a las personas un método único de percibir y actuar con el entorno que nos rodea (Culque Toapanta, 2022).

### **1.1.1 Interfaces Hápticas**

Una interfaz háptica es un dispositivo que permite la interacción con el entorno virtual o sistema operativo remoto, utilizada para tareas habituales como la manipulación en el mundo real de objetos (Abdi, 2020).

En la Figura 2 se muestra que a través de una acción que realiza una persona, se generará una fuerza de reacción para que el cuerpo del usuario la interprete como información táctil. Mediante la simulación de dicha acción de forma virtual e interpretación de los sensores biomecánicos.

**Figura. 2**  
*Interfaz Háptica*



**Fuente.** Tomado de (Abdi, 2020)

Los métodos hápticos de varios dedos proporcionan una interfaz humano-máquina realista y natural que mejora la comunicación cuando se interactúa con entornos virtuales.

Entre estos se encuentran los guantes hápticos que pueden simular sensaciones reales basadas en interacciones virtuales, permitiendo al usuario captar y sentir objetos virtuales, como si estos fueran reales. A su vez, debido a la alta destreza manual que pueden proporcionar estos dispositivos se puede aplicar para controlar el movimiento de robots remotos (Jorge López, 2019).

### 1.1.2 Sistemas Hápticos

El sistema háptico interactúa con el usuario mediante medios de interacción táctil. También conocido como dispositivo háptico, el sistema háptico incluye no solo entradas y salidas básicas, sino también el sistema de control necesario para controlar actuadores, sensores y procesar información (Peña Cortés, 2022).

Las interfaces hápticas no solo conectan la percepción del usuario con funciones de activación, sino también con funciones sensoriales dedicadas para capturar las intenciones

del usuario, con la finalidad de comunicar dichas interacciones en entornos reales o virtuales (Peña Cortés, 2022).

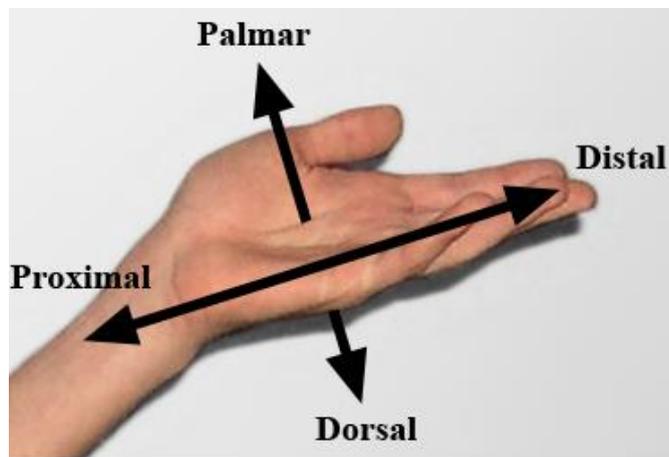
### 1.1.3 Biomecánica de la mano

La biomecánica de la mano combina el características de la ingeniería mecánica con la biología y la fisiología humana, hace referencia a los movimientos normales de las articulaciones que en este caso hablaríamos de los movimientos de la mano que son: inversión, eversión, flexión y extensión de muñeca, presión palmar, flexión, extensión, aducción y abducción de dedos.

En esencia, la biomecánica es una ciencia multidisciplinaria que se encarga de la aplicación de principios mecánicos al cuerpo humano en movimiento y reposo (Sánchez J. , 2017).

La mano humana es uno de los principales órganos para poder interactuar con el entorno. Cada mano contiene aproximadamente 29 huesos y 34 músculos que permiten mover todos los dedos. Tales grados de libertad, y conexiones neuronales masivas, proveen una enorme flexibilidad para realizar todo tipo de movimiento de los dedos, tales como tocar, presionar, agarrar, apretar, tirar, entre otros (Rodríguez Ramirez, Ruiz, & Torres, 2020). En la Fig. 3 podemos observar los lados de la mano.

**Figura. 3**  
*Axisas de la mano*



**Fuente.** Tomado de (Esseh)

Para lograr el funcionamiento completo de la mano, o incluso la capacidad mínima de agarre, se requiere que el pulgar trabaje en combinación con el dedo índice. La oponibilidad

del pulgar y su fuerza es fundamental para la interacción con las manos y la manipulación de objetos. Se considera comúnmente que el pulgar es el responsable de al menos el 50 % de la función de movimiento de la mano. A partir de la muñeca hasta la punta del pulgar, hay 4 huesos que son el trapecio, el metacarpiano, la falange proximal y la falange distal, estas tres articulaciones del pulgar son: la coyuntura carpometacarpiana, coyuntura metacarpofalángica y la coyuntura interfalángica (Escobar & Quiros, 2017). En la Figura 4 podemos observar los huesos a detalle de la mano que nos servirán para la composición y construcción del exoesqueleto.

**Figura. 4**

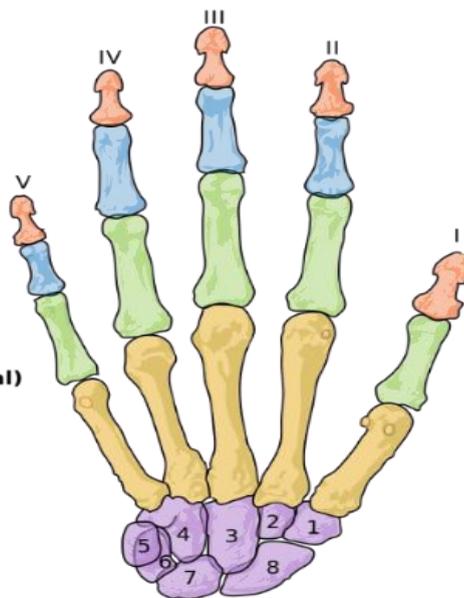
*Huesos de la mano*

- Falanges distales (3ª)
- Falanges medias (2ª)
- Falanges proximales (1ª)
- Metacarpianos o metacarpos
- Carpianos o carpos

### Huesos de la mano izquierda

**Visión posterior (dorsal)**

1. Trapecio
2. Trapezoide
3. Grande
4. Ganchoso
5. Pisiforme
6. Piramidal
7. Semilunar
8. Escafoides



**Fuente.** Tomado de (Mendez A. Y., 2012)

Desde la muñeca hasta la punta del pulgar, existen cuatro huesos: el trapecio (hueso carpiano), el primer metacarpiano, la falange proximal y la falange distal. Las tres articulaciones del pulgar son: articulación carpometacarpiana (CMC), articulación metacarpo falángica (MCP) y articulación interfalángica (IP) (Berezowsky, 2018). En la Figura 5 podemos ver los huesos y articulaciones competentes al dedo pulgar humano.

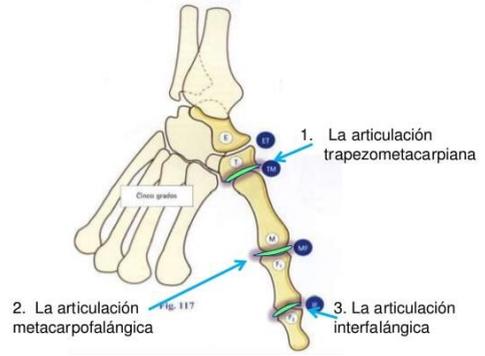
**Figura. 5**

*Articulaciones y huesos del pulgar*

**HUESOS**



**ARTICULACIONES**



**Fuente.** Tomado de (Berezowsky, 2018)

## 1.2 Guantes con tipos de retroalimentación

Existen dos tipos de guantes usados en su mayoría para realidad virtual ya sea para videojuegos proyectos electrónicos o prótesis entre ellos hay dos tipos de retroalimentación que son de fuerza y vibratoria.

### a) Retroalimentación vibratoria

El guante construido, llamado Virtual Touch, involucra sensores de flexión con el fin de sentir en todo momento el movimiento que acude en la mano, también motores pequeños de vibración que informan al usuario el contacto virtual con un objeto. Se implementaron dos tipos de software para manejar el dispositivo, primero que todo una aplicación en el ambiente ROS basado en Linux, y como segunda instancia una aplicación en Windows utilizando como motor de renderizado gráfico a VTK (Vivas, 2018). En la Figura 6 podemos ver un ejemplo de guante de retroalimentación vibratoria.

**Figura. 6**

*Guante con retroalimentación de vibración*



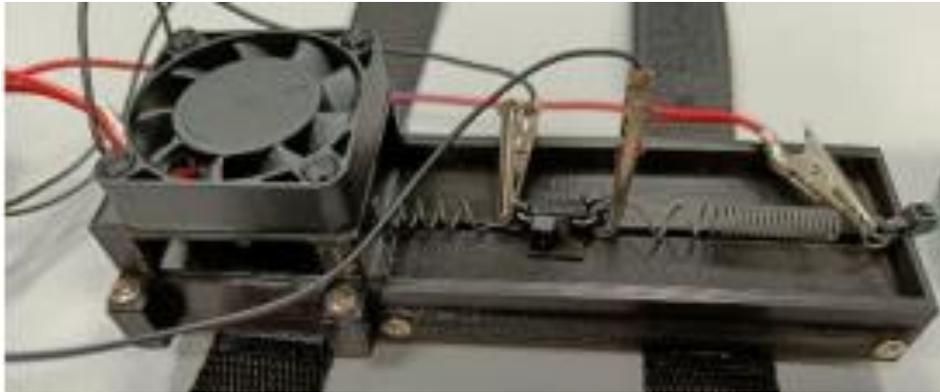
**Fuente.** Tomado de (Vivas, 2018)

## b) Retroalimentación de fuerza

Un prototipo llamado SMA fabrico un prototipo de guante con retroalimentación de fuerza usando un muelle para estirar y contraer esto funcionaria de la misma forma como los servos, ya que cuando el guante siente algo el muelle se contrae dando una fuerza de retracción la que hace que el guante no pueda cerrarse hasta cierto límite como se observa en la Figura 7 (Mendez, 2018).

### Figura. 7

*Guante con retroalimentación de fuerza*



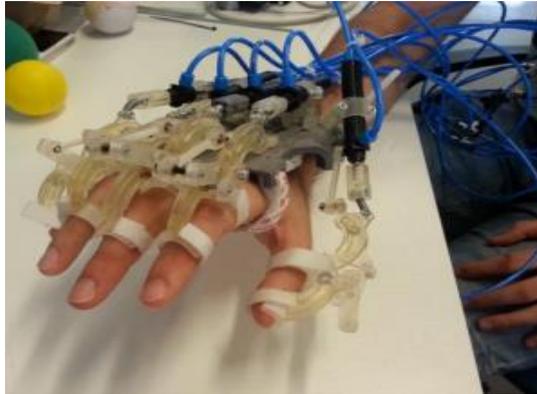
**Fuente.** Tomado de (Mendez, 2018)

## 1.3 Guante de tipo Exoesqueleto

En la háptica es muy común la utilización de exoesqueletos, ya que permiten un mayor grado de interacción y un menor margen de error al momento de ejercer un movimiento en específico, por lo que cada grado de torción de una articulación nos puede definir un valor determinado para ejercer una acción con respecto a cada hueso del esqueleto óseo del usuario (du Plessis, 2021). A continuación, en la Figura 8 se hace la representación de un exoesqueleto para una mano de un adulto promedio.

## **Figura. 8**

*Exoesqueleto de una mano*



**Fuente.** Tomado de (S.C. Enriquez, 2014)

### **1.4 Sistemas de Control Háptico**

El sistema de control háptico es la parte del sistema real que controla los componentes mecánicos y eléctricos del dispositivo para una adecuada detección, manipulación y despliegue de información háptica, si no también de la conexión con otras partes del sistema háptico. Lo cual puede ser, la unión entre la interfaz háptica y el manipulador (HMI) (Valenzuela Urrutia, 2016).

### **1.5 Hardware de Control**

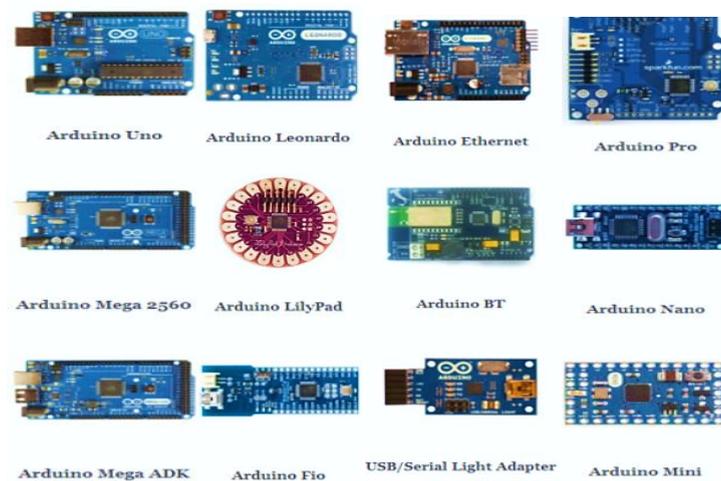
La ciencia en la humanidad cada día se propone en facilitar vida de las personas constituyendo entre ellos a una gran variedad de controladores y módulos automatizados que facilitan y tienen varias aplicaciones de uso en la industria y en cuanto a uso personal. Entre los dispositivos o Hardware de control más usados en la actualidad están: Los PLC, módulos Arduino, módulos de control de potencia, y una extensa variedad de sensores y actuadores acoplables al sistema que facilitan el manejo y control de las diversas aplicaciones.

#### **1.5.1 Tarjeta de desarrollo**

Esta placa de desarrollo está basada en hardware libre, este dispositivo dispone de un microcontrolador reprogramable que permite tener comunicación alámbrica o inalámbricamente dependiendo los dispositivos añadidos al sistema. Arduino es una empresa que dispone de varias y diferentes placas dependiendo la necesidad del usuario (Fernandez, 2020). En la Figura 9 se observa algunos de los módulos Arduino existentes.

## Figura. 9

### Módulos Arduino



Fuente. Tomado de (Ramar, 2014)

De los cuales los más utilizados suelen ser el Arduino Nano, Arduino Mega, Arduino Uno, Arduino Due, Arduino Ethernet, Wemos y como última aportación en cuanto flexibilidad, diseño y tamaño el ESP-32.

#### 1.5.1.1 Arduino Mega

Esta placa de desarrollo se basa en un microcontrolador ATmega2560 y está diseñada para proyectos que requieren una variedad de entradas y salidas analógicas y digitales, ya que este es una de las placas con más entradas digitales. Por lo tanto, se utiliza en procesos más complejos. El Arduino Mega ya viene cargado un gestor de arranque lo que permite cargar códigos sin necesidad de un programador por hardware externo. (Arduino, 2022).

#### 1.5.2 Sensor Infrarrojo de distancia marca Sharp

Los sensores infrarrojos Sharp funcionan mediante el rebote de un haz infrarrojo, este rebota en el objeto y es captado por un array de fototransistor infrarrojo. Para estos sensores hay variedad de distancias que manejan que se definen más adelante en la Tabla 2. (Mera & Ruano, 2019). En la Figura 10 observaremos la estructura común de un Sensor Sharp que se encuentra comercialmente.

**Figura. 10**

*Sensor de distancia Sharp*



**Fuente.** Tomado de (Mera & Ruano, 2019)

Este tipo de sensor mide la distancia entre el objeto y él, haciendo uso de la triangulación por medio de su emisor infrarrojo y su receptor y la detección de tipo reflectivo.

**Tabla 2**

*Modelos de Sensores Infrarrojos según su distancia*

| MODELO              | RANGO DE MEDICIÓN |
|---------------------|-------------------|
| <b>GP2Y0A41SK0F</b> | 4cm-30cm          |
| <b>GP2Y0A21YK0F</b> | 10cm-80cm         |
| <b>GP2Y0A02YK0F</b> | 20cm-150cm        |

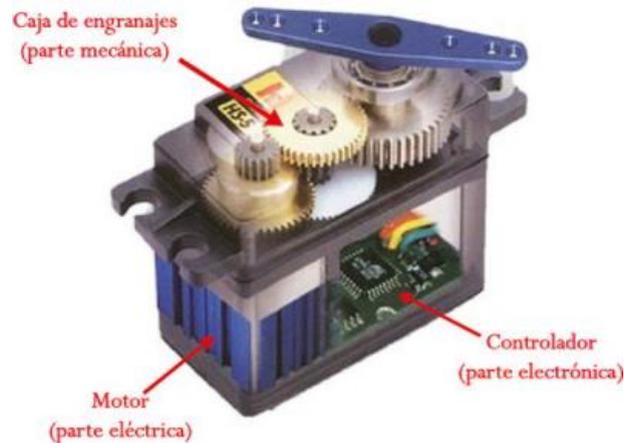
**Fuente.** Tomado de (ElectroStore, 2022)

### 1.5.3 Servomotor

Es un motor que puede controlar con precisión la aceleración y la velocidad, características que no tienen los motores convencionales. Utiliza un motor convencional y lo combina con un sensor de retroalimentación de posición. Se controlan enviando pulsos eléctricos de ancho variable, o modulación de ancho de pulso (PWM), a través de la línea de control. Hay un pulso mínimo, un pulso máximo y una frecuencia de repetición (Salazar, 2020). En la Figura 11 se observa los engranajes de un servomotor y el controlador del motor.

Es un motor con sistema de circuito cerrado para un control preciso en cuanto a aceleración y velocidad, mediante la utilización de pulsos eléctricos o PWM para su funcionalidad. Dicho circuito cerrado consta de un motor normal y un sensor para su retroalimentación con respecto a posición (Salazar, 2020). En la Figura 11 se observa los engranajes de un servomotor y el controlador del motor.

**Figura. 11**  
Servomotor



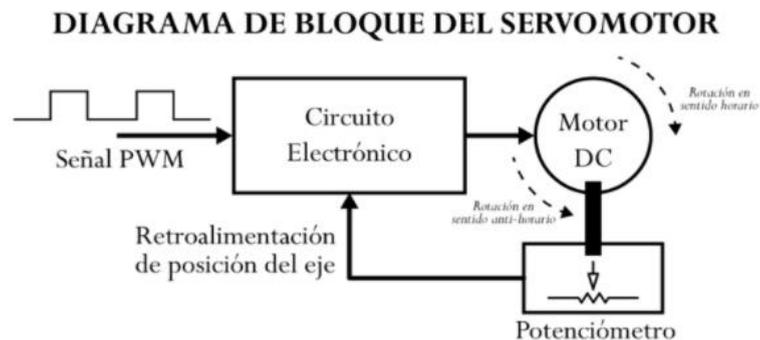
**Fuente.** Tomado de (Kadhum, 2021)

El diagrama de bloque del servomotor representa de forma visual el servomotor como un sistema. Los circuitos electrónicos son responsables de recibir la señal PWM y convertirla en movimiento de motor de CC. El eje del motor de CC está conectado a un potenciómetro que puede formar un divisor de voltaje. El voltaje en la salida del divisor varía en función de la posición del eje del motor DC (Kadhum, 2021).

En la Figura 12 se observa ver el diagrama de bloque del servomotor el cual envía una señal PWM al circuito electrónico para hacer girar al motor mediante un potenciómetro.

**Figura. 12**

*Diagrama de bloques del servomotor*



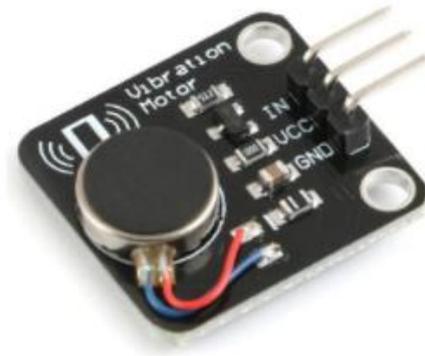
**Fuente.** Tomado de (Kadhum, 2021)

#### 1.5.4 PWM Vibrador Arduino

Este tipo de motor se asemeja a un vibrador de teléfono móvil siendo amplificado, en donde su intensidad de vibración del motor puede ser controlada por PWM. El módulo puede convertir la señal eléctrica a vibración mecánica. Cuando la entrada es alta, el motor vibra, al igual que tu teléfono en modo silencioso, teniendo una tensión nominal de 5V (Perez, 2020). En la Figura 13 se observa el módulo vibrador compatible con Arduino.

**Figura. 13**

*Modulo vibrador PWM*



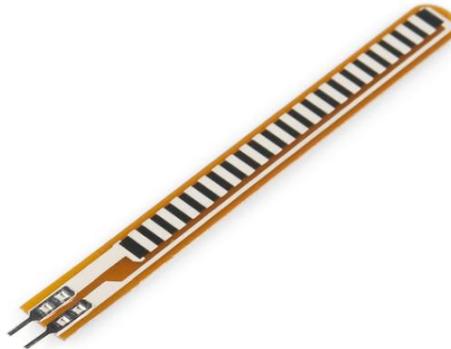
**Fuente.** Tomado de (Perez, 2020)

#### 1.5.5 Sensor Flexométrico

Este tipo de sensor es muy útil, ya que su resistividad cambia al ser flexionado, por lo que lo vuelve idóneo para medir el ángulo de giro en articulaciones en cuanto a aplicaciones robóticas, biométricas o de interfaz de usuario.

Este tipo de sensor está diseñado para hacer uso de su flexión en una sola dirección, con los contactos al exterior. Se debe tener precaución de no flexionar el sensor en la sección de la base, ya que al hacerlo puede ocasionar ruptura del material conductor por su sensibilidad (AV Electronics, 2019). En la Figura 14 podemos ver los sensores flexométricos sensibles.

**Figura. 14**  
*Sensores Flexométricos*



**Fuente.** Tomado de (*AV Electronics, 2019*)

### 1.5.6 Motorreductores DC

Estos motorreductores son los que provocan movimiento en dispositivos robóticos, ya que en un sistema de control debe reaccionar a las señales o acciones emitidas por el usuario. En este tipo de aplicaciones lo que se busca son motores pequeños para que no ocupen mucho espacio, livianos y con fuerza, ya que cuenta con un conjunto de engranajes que actúan como reductores de velocidad, pero aumento de torque, en la Figura 15 se muestra un motorreductor Pololu (*Mena Palomeque, 2017*).

**Figura. 15**  
*Motorreductor Pololu*



**Fuente.** Tomado de (*Mena Palomeque, 2017*)

Los motorreductores Pololu cuentan con sistemas de relación de reducción: que es 10:1, 50:1, 100:1, 150:1 lo que significa que a menos valor de reducción aumenta la velocidad, pero

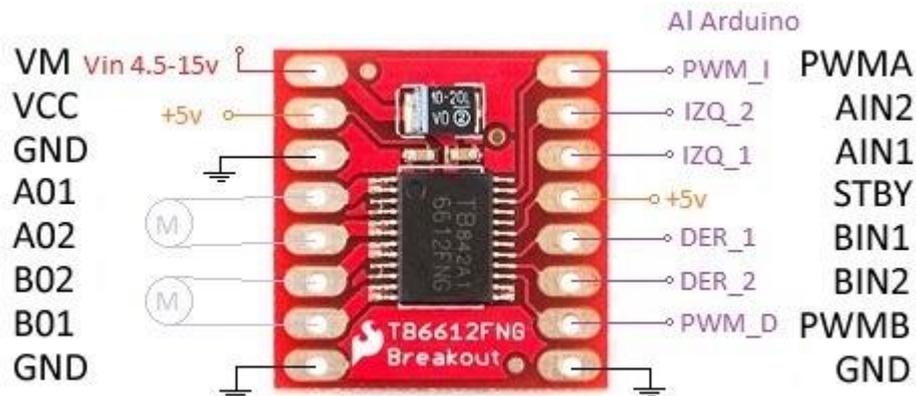
se pierde torque y viceversa en estos casos la velocidad que se obtiene es 3100, 590, 310, 210 revoluciones por minuto (Robotics Sando, 2022).

### 1.5.7 Driver TB6612fNG

Dado que el Arduino no puede entregar directamente la potencia que necesitan los motores, se utilizan un driver para comandar los motores. En donde los drivers envían las señales digitales para indicar en qué dirección debe girar el motor y suministrar la potencia necesaria al motor, en la Figura 16 podemos ver el módulo controlador para potencia de motores reductores (Enriquez, 2018).

**Figura. 16**

*Driver TB6612fNG*



**Fuente.** Tomado de (Enriquez, 2018)

### 1.5.8 Batería Li-Po

Esta batería es recargable siendo un dispositivo de almacenamiento y suministro de energía eléctrica, tiene la capacidad de almacenar altas densidades de energía gracias a su componente químico, que es el Polímero de Litio. Estas baterías recargables son usadas en aplicaciones que requieren corrientes superiores a 1A con bajo peso y tamaño reducido, hay que destacar que algunos elementos no soportan el voltaje de tus baterías, necesitas una PDB que reciba la corriente de tu batería y la convierta a los voltajes que tus componentes soporten (Islas Silvas, 2019). En la Figura 17 se observa una batería tipo Li-Po y la PBD.

## Figura. 17

Batería tipo Li-Po y placa distribuidora de voltaje PDB



Fuente. Tomado de (Techmake, 2018)

### 1.5.9 Módulo de carga BMS 3S 10A

Esta placa de protección de baterías de litio cuenta con una función de balanceador y recuperación automática. Maneja un voltaje de 12.6V y una corriente máxima de 10A, por lo que una de sus grandes ventajas es que puede cargar baterías de 3.6V o 3.7V como las 18650 o las 26650 (Sando Robotics, 2020).

## 1.6 Módulos de comunicación

Existen varios tipos de comunicación inalámbrica permitiendo enviar y recibir información de manera rápida y segura algunas tecnologías tienen un alcance mayor que superan los 1000 metros de las cuales podemos definir la siguiente:

### 1.6.1 Módulo LoRa

Estos módulos fueron creados e impulsados por LoRa Alliance Association, que se ocupa de la comunicación IoT, M2M, ciudades inteligentes y más. Los principales fabricantes de dispositivos LoRa son SETMech y Microchip Corporation. Esta es una red inalámbrica que utiliza una tecnología de capa física de secuencia directa de espectro ensanchado. Permite comunicarse en señales con un valor bajo de potencia y utiliza modulación por desplazamiento de frecuencia (Narvaez & Contreras, 2020). En la Figura 18 se observa algunos módulos de comunicación LoRa.

**Figura. 18**

*Tipos de módulos LoRa*

| Model No.             | RFM300W (TRX)   | RFM300HW (TRX)  | RFM63W (TRX)   | RFM64W (TRX)  |
|-----------------------|---|---|--|---|
| Photo                 |  |  |  |  |
| Frequency(MHz)        | 315/433/868/915   | 315/433/868/915   | 868/915  | 315/433   |
| VDD(V)                | 1.8-3.6   | 1.8-3.6   | 2.1-3.6  | 2.1-3.6   |
| Modulation            | (G)FSK/OOK  | (G)FSK/OOK  | FSK/OOK  | FSK/OOK   |
| Max. Data Rate (kbps) | 300/40  | 300/40  | 200  | 200   |
| Current (mA)          | Tx  | 28  | 75   | 25  |
|                       | Rx  | 7   | 7  | 3   |
| Sensitivity (dBm)     | -120  | -120  | -110   | -110  |
| Output Power (dBm)    | 13  | 20  | 12   | 12  |
| Size(mm)              | 16x16x1.9   | 16x16x1.9   | 19.7x16x1.9  | 19.7x16x1.9   |
| Interface             | SPI   | SPI   | SPI  | SPI   |

**Fuente.** Tomado de (Hoperf, 2022)

Como se observa en la Figura 18 existen varios tipos de módulos de comunicación, la palabra LoRa significa (Long Range) Larga distancia. Esta es una tecnología de radiofrecuencia que puede utilizarse para transmitir información bidireccional a largas distancias sin consumir demasiada energía. Los sensores remotos pueden usar este atributo y solo necesitan operar una pequeña batería para transmitir datos, la tasa de datos está entre 0.3 a 50 Kbits/s por canal y su alcance de distancia es de 15 kilómetros (Narvaez & Contreras, 2020).

### 1.6.2 Dragino Shield LoRa 433 MHz

Este Shield bidireccional para Arduino está basado en el chip FRM98W funcionando a 433 MHz que permite transmitir datos a larga distancia con una mínima pérdida, con muy bajo ruido y bajo consumo, teniendo una corriente de 10.3 mA en transmisión de datos y 200nA en modo reposo (WAVGAT, 2022). En la Figura 19 se observa el Shield LoRa 433MHz.

**Figura. 19**

*Dragino Shield LoRa 433MHz*



**Fuente:** Tomado de (WAVGAT, 2022)

### 1.6.3 Giroscopio

Un giroscopio es un instrumento que permite medir, mantener y hasta modificar la dirección en el espacio de un objeto o vehículo. Este dispositivo tiene la capacidad de girar sobre un eje y al ser sometido a una fuerza que debería hacer que caiga, este se mantiene girando sin detenerse (Rodríguez L. , 2022).

El módulo giroscopio está basado en el sensor MPU6050 y contiene todo lo necesario medir movimiento en 6 grados de libertad, combinando un giroscopio de 3 ejes y un acelerómetro de 3 ejes en un mismo chip, El acelerómetro permite detectar la orientación y rotar en función de esta (Rodríguez L. , 2022). En la Figura 20 se demuestra el giroscopio MPU6050.

**Figura. 20**

*MPU6050*



**Fuente.** Tomado de (Naylamp, 2022)

Se comunica a través de una interfaz I2C y posee una librería muy difundida para su uso inmediato. Su conexión es sencilla a través de su interfaz I2C máster, permitiendo así controlar sensores externos adicionales como magnetómetros o barómetros, entre otros, sin intervención del procesador principal (Naylamp, 2022).

Para una captación precisa del movimiento, posee un rango programable de escala de 250/500/1000/2000 grados/seg. Para el giroscopio y de 2g/4g/8g/16g para el acelerómetro (Naylamp, 2022).

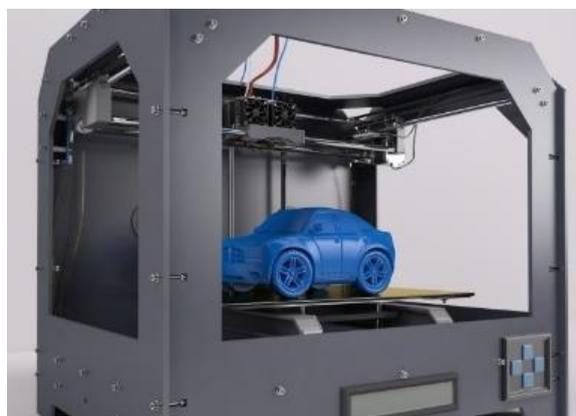
### 1.7 Servicio de impresión 3D

Las impresoras 3D son dispositivos capaces de emitir e imprimir figuras en volumen diseñadas por un computador, pasando de un diseño en tercera dimensión virtual a un físico, también las impresoras 3D imprime objetos a diferentes escalas y materiales de construcción, el método de impresión que se usa comúnmente es uniendo material mediante capas y patrones (Dominguez, 2017).

El proceso de impresión tarda dependiendo de la complejidad y el tamaño de la pieza. En la Figura 21 se observa un modelo estándar de una impresora 3D el cual se usa para fabricación de piezas las que son diseñadas para proyectos, ya que son diseños propios de personas dedicadas al área de experimentación o implementación de prototipos, proyectos innovativos en el área de la electrónica.

**Figura. 21**

*Impresora 3D*



**Fuente.** Tomado de (Teroson, 2018)

## 1.8 Software de Desarrollo

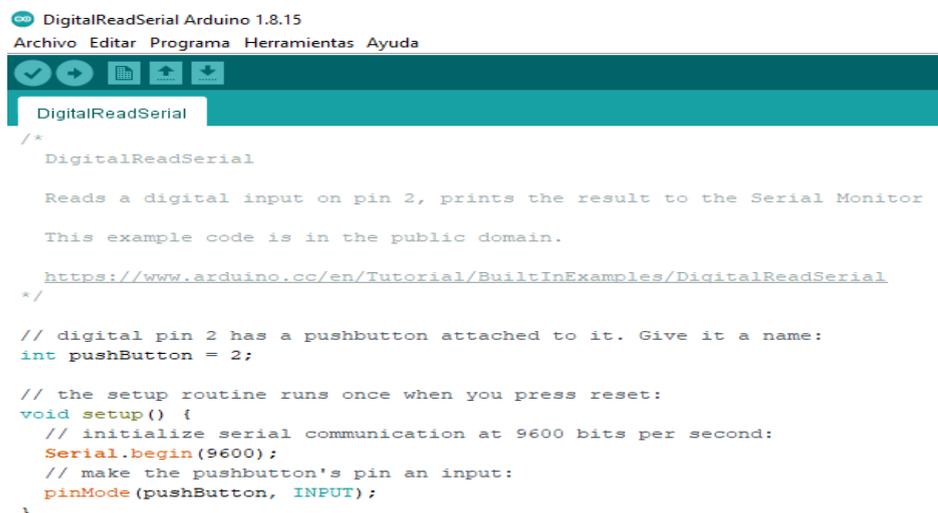
La plataforma de Arduino es un software de uso libre y con gran número de librerías diseñadas para distintas aplicaciones por lo que hace posible trabajar con este software para un guante aptico con retroalimentación de fuerza.

### 1.8.1 Arduino IDE

Es una plataforma de creación de prototipos electrónicos de código abierto basada en hardware y software de entorno de procesamiento y un lenguaje de programación basado en cables. Dado que Arduino es una computadora pequeña, ejecuta una serie de códigos ingresados por el usuario desde la computadora.

**Figura. 22**

*Interfaz Arduino IDE*

The image shows a screenshot of the Arduino IDE interface. At the top, the title bar reads "DigitalReadSerial Arduino 1.8.15". Below the title bar is a menu bar with "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". A toolbar with icons for check, back, forward, and refresh is visible. The main editor area displays the code for the "DigitalReadSerial" example. The code includes a multi-line comment describing the example, a URL to the Arduino website, and C++ code for pin configuration and serial communication.

```
DigitalReadSerial
/*
  DigitalReadSerial

  Reads a digital input on pin 2, prints the result to the Serial Monitor

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/DigitalReadSerial
*/

// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 2;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
}
```

**Fuente. Autor**

Este programa es denominado IDE, el cual significa "Integrated Development Environment" ("Entorno de Desarrollo Integrado"). Este debe estar instalado en un ordenador, su interfaz es muy amigable y sencilla de usar, para que Arduino ejecute las acciones solicitadas. Una vez escrito, se carga a través del USB y el Arduino comenzará a trabajar de forma autónoma (León, 2019). En la Figura 22 podemos observar la interfaz de Arduino IDE donde insertamos secciones de acción y comentarios para una mayor comprensión del programa a comandar.

La información adquirida en este capítulo determinó que existen tecnologías que trabajan conjuntamente para un fin común, mediante el análisis de cada componente recopilado se obtuvo que por medio de las distintas tecnologías existentes se puede elaborar y sistematizar guantes hápticos acorde a los requerimientos y características a solventar, para ello se tomará en cuenta la información obtenida mediante el análisis del hardware detallado permitiendo una selección de componentes y un diseño para la implementación del prototipo final.

## CAPÍTULO 2

### Diseño del guante háptico con retroalimentación de fuerza

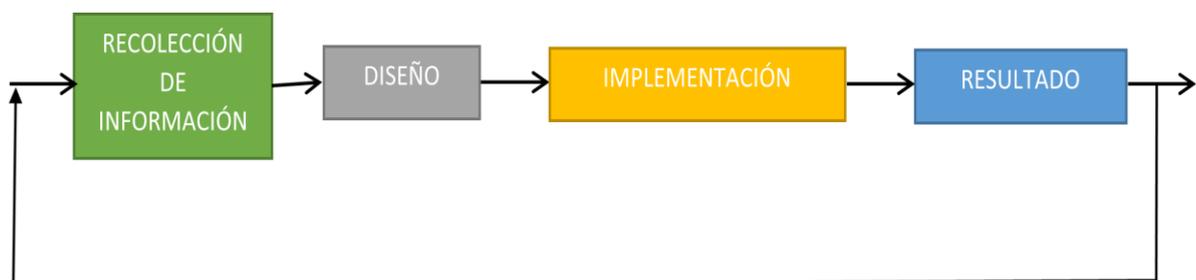
A continuación, se presentan los distintos elementos electrónicos que mediante el análisis fueron seleccionados para el diseño del guante háptico, la metodología que se utilizó para la selección de los materiales para el cumplimiento de las distintas prestaciones que requiere el guante háptico, con la finalidad de cumplir los objetivos propuestos.

#### 2.1 Metodología

Para el progreso de este capítulo se usó la metodología en espiral la cual permite seguir un proceso ordenado e interactuar con los anteriores procesos para obtener los mejores resultados del prototipo, con el fin de evitar que se presenten riesgos que impidan el correcto funcionamiento del guante háptico y el vehículo para ello se detalla la secuencia que se seguirá acorde a la metodología propuesta como se observa en la Figura 23.

**Figura. 23**

*Diagrama de bloques de la metodología a seguir*



**Fuente.** Autor

En la primera sección, se hizo una recolección de información sobre los componentes electrónicos que existen para cumplir la necesidad del prototipo, en la segunda fase se realizó el diseño, ya que una vez seleccionados los componentes ya se tiene claro los aspectos físicos que tiene el guante háptico, como tercera fase es la implementación y montaje de todos los componentes y como última fase los resultados para realizar las pruebas y ajustes del prototipo.

#### 2.2 Características de guante háptico

Para la realización del prototipo se tuvo en cuenta algunas características para cumplir con los requisitos mínimos que cumple un guante háptico las cuales son las siguientes:

- Bajo costo
- Robustez
- Fácil adquisición de materiales
- Fácil operación
- Buena comunicación
- Fluidez y respuesta
- Tamaño promedio de una persona adulta
- Guante para la mano derecha
- Tiempo continuo de uso

En cuanto a la elección de la mano, se basó en el estudio de la revista Business Insider donde (Priego, 2022) determino que el 90% de personas existentes en la población son diestros y solo un 10% son zurdos, por ello se optó a implementarlo todo en la mano derecha

### **2.3 Tipo de guante háptico**

Para la elaboración del guante háptico se optó por montar el exoesqueleto en el dorso de la mano, y la parte que controla al exoesqueleto se colocara en el antebrazo, en la que se realiza todas las conexiones y alimentación del mismo para proteger el controlador y sus respectivos dispositivos electrónicos, para ello se tomó el diseño de la Figura 24 como base de diseño en la que se modificó algunas partes para el funcionamiento requerido.

**Figura. 24**

*Imagen referencial del guante háptico*

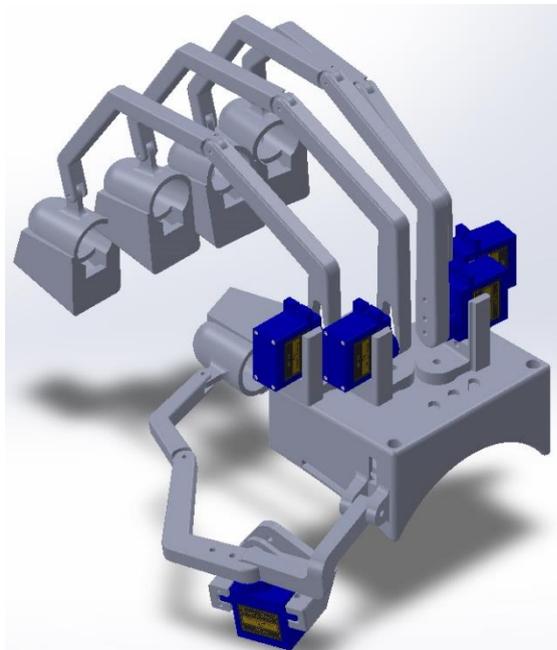


**Fuente.** Tomada de (Curtis, 2021)

Este diseño de exoesqueleto (Curtis, 2021), se tomó como referencia para proceder a modificarlo a conveniencia del prototipo a realizar, con el fin de que cumpla un correcto funcionamiento y se obtuvo el siguiente resultado que se muestra en la Figura 25.

**Figura. 25**

*Exoesqueleto guante háptico*

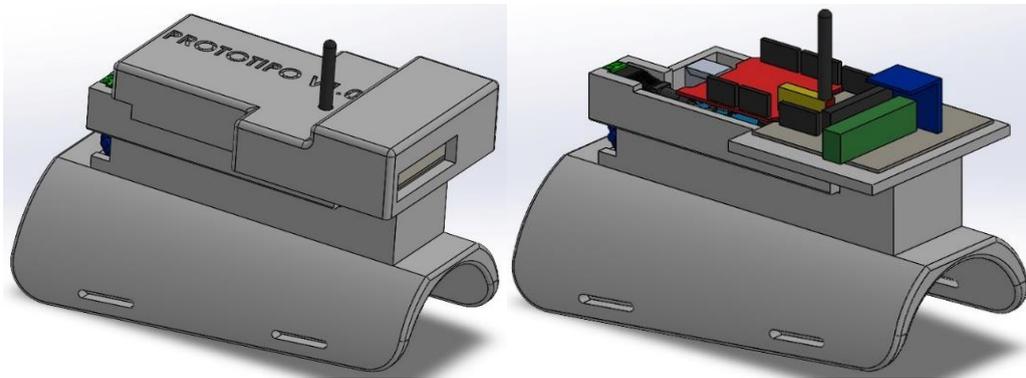


**Fuente.** Autor

En la parte de cada dedo se realizó un compartimento para colocar los módulos motor-vibradores y los servomotores controlan el movimiento de cada dedo del exoesqueleto, ya sea para abrir o cerrar la mano. Para la parte del antebrazo se realizó el diseño de un compartimento donde aloja a todos los componentes del guante háptico con el fin de proteger y asegurar a los elementos electrónicos como se observa en la Figura 26.

**Figura. 26**

*Exoesqueleto ante brazo guante háptico*

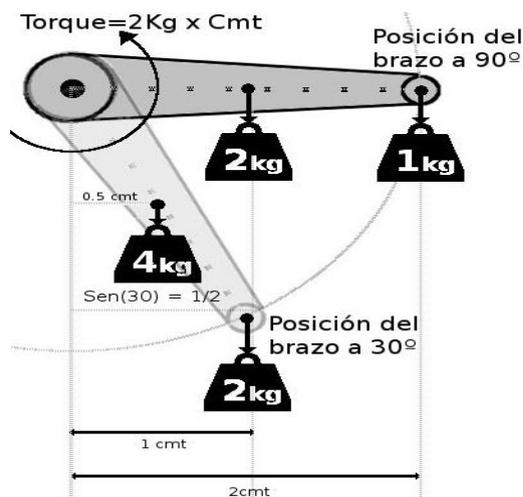


**Fuente.** Autor

Este diseño es impreso en 3D haciendo uso del diseño referenciales para adaptar al prototipo requerido. Para el cálculo del servomotor se utilizó la ejemplificación de la Figura 27.

**Figura. 27**

*Valor del torque según el brazo del eje del servo*



**Fuente.** Tomado de (Castro, 2019)

Como se observó en la Figura 27, dependiendo de largo del brazo del eje del servo y su ángulo de inclinación varía el peso que puede soportar basándose en esto se usó un brazo de 2 centímetros y un ángulo de 60° como máximo con el fin de que abra y cierre el guante, para ello se utiliza la Ecuación 1 (Castro, 2019).

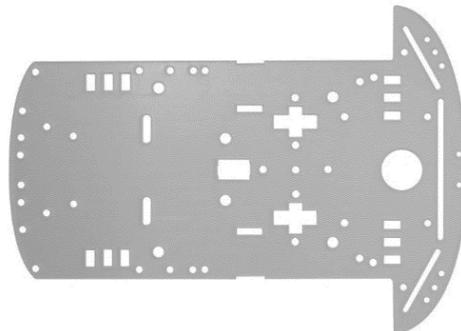
$$Peso\ max = peso\ servo * centimetros * seno(\acute{a}ngulo) \quad \text{Ecuación 1}$$

## 2.4 Tipo de vehículo

Para montar todos los componentes electrónicos del vehículo se optó utilizar el chasis de una base de aluminio al cual se le acoplará para colocar los componentes que integran el vehículo, como se observa en la Figura 28 es un diseño referencial para detallar las características del vehículo, para lo cual este chasis soportará todo el peso de los elementos juntamente con las llantas de goma.

**Figura. 28**

*Imagen referencial del vehículo*



**Fuente.** Tomado de (Imrelevsa, 2022)

Este vehículo es para uso único del guante háptico como método de comprobación de la velocidad de respuesta entre el guante háptico y el vehículo por lo que su función es para pruebas del prototipo, por lo que se detalla las características del vehículo en la Tabla 3.

**Tabla 3**

*Características del vehículo*

| Características | Detalles  |
|-----------------|---|
| Tipo de uso     | El uso del vehículo se utilizó para ejemplificar la respuesta de retroalimentación del guante háptico, ya |

|                         |   |
|-------------------------|---|
|                         | que ambos se comunican entre sí de una forma bidireccional.   |
| <b>Tipo de vehículo</b> | El tipo de vehículo es de 3 ruedas, donde sus dos ruedas traseras son las que proporcionan la fuerza para mover el vehículo por medio de dos motores y la rueda delanteras brinda la dirección de giro del vehículo por medio de un servomotor. |
| <b>Terreno</b>          | El vehículo se usó para superficies planas regulares ya que solo recibe señales del guante como método de comprobación y solo se usa para distancias cortas, por lo que no es necesario hacer un vehículo a escala.                             |

*Fuente. Autor*

## 2.5 Componentes que integran el guante háptico

Para los componentes del guante háptico se realizó una comparación entre tecnologías en cuanto a controladores sensores y actuadores con el fin de detallar los componentes que se utilizan para la implementación y usar una batería acorde al uso que se vaya a utilizar.

### 2.5.1 Controlador para el guante háptico

A continuación, se detalla e informa las características de los controladores Arduino, ya que es de la familia de dispositivos programables electrónicos, para lo que se tuvo en cuenta varias características como velocidad de procesamiento, precio y rendimiento, por lo que se detalla en la Tabla 4.

**Tabla 4**

*Características de la familia de tecnologías de Arduino*

| Arduino      | Entradas y salidas digitales | Entradas y salidas analógicas | Voltaje (v) | Corriente (mA) | Voltaje de operación (v) | Clock Speed (MHz) | Peso (g) |
|--------------|------------------------------|-------------------------------|-------------|----------------|--------------------------|-------------------|----------|
| <b>Due</b>   | 54                           | 12                            | 7 - 12      | 200            | 3.3                      | 84                | 36       |
| <b>Mega</b>  | 54                           | 12                            | 7 - 12      | 200            | 3.3 – 5                  | 16                | 37       |
| <b>Uno</b>   | 14                           | 6                             | 7 - 12      | 60             | 3.3 – 5                  | 16                | 25       |
| <b>Nano</b>  | 22                           | 8                             | 7 - 12      | 19             | 3.3 – 5                  | 16                | 7        |
| <b>Esp32</b> | 16                           | 2                             | 3.3 - 5     | 19             | 3.3                      | 16                | 12       |

**Fuente.** Tomado de (Roda, 2020), (Llamas, 2017), (Meier, 2020)

Con la Tabla 4 de características técnicas todos los controladores Arduino funcionan de 7 a 12 voltios a excepción del ESP32 para lo cual se optó por el Arduino Mega, ya que cuenta con varias entradas analógicas y digitales, ya que el guante requiere varias entradas análogas además de tener pines de conexión lo que evita que se realice una placa de expansión adicional para poder utilizar los pines.

### 2.5.2 Sensores para el guante háptico

Se realizó un análisis entre los dos tipos de sensores Flex los que son sensores resistivos que se detallan las características en la Tabla 5.

**Tabla 5**  
*Características de sensores*

| Sensor Flex       | Voltaje (v) | Corriente (mA) | Curva de resistencia (kΩ) |
|-------------------|-------------|----------------|---------------------------|
| 4.5 pul. (11.5cm) | 5 - 12      | 20             | 60 – 110                  |
| 2.2 pul. (4.5 cm) | 5 - 12      | 10             | 45 - 145                  |

**Fuente.** Tomada de (Rambal, 2022)

Para el guante háptico se utilizó el sensor Flex de 4.5 pulgadas, ya que el sensor Flex de 2.2 pulgadas es muy pequeño, lo que dificultaría al momento de ensamblaje en el guante que ocupe todo el espacio de los dedos y sobre los conectores se verán afectados por el movimiento de los dedos del guante.

### 2.5.3 Comunicación para el guante háptico

Se detalló 2 tecnologías de comunicación mediante Bluetooth y radiofrecuencia indicando varias características cómo se proporciona en la Tabla 6.

**Tabla 6**  
*Características de los módulos de comunicación*

| Tecnologías | Módulos | Frecuencias | Alcance línea de vista (m) | Voltaje (v) | Corriente (mA) |
|-------------|---------|-------------|----------------------------|-------------|----------------|
| Bluetooth   | HC-05   | 2.4 GHz     | 10                         | 3.5         | 50             |
|             | HC-06   | 2.4 GHz     | 10                         | 3.5         | 40             |

|             |                             |         |           |         |     |
|-------------|-----------------------------|---------|-----------|---------|-----|
| <b>LoRa</b> | Shield LoRa<br>para Arduino | 433 MHz | 172 - 565 | 3.3 - 5 | 150 |
|-------------|-----------------------------|---------|-----------|---------|-----|

**Fuente.** Tomado de (Mera & Ruano, 2019), (Rodríguez D. , 2018)

Para realizar el guante háptico se utilizó el módulo Shield LoRa, ya que se acopla fácilmente para trabajar con Arduino Mega y Due, ofrece una comunicación a larga distancias y el consumo de corriente es baja.

#### 2.5.4 Actuadores o motores para el guante háptico

A continuación, se detalla en la Tabla 7 las características del servomotor y motor vibrador en el cual es un módulo compatible con Arduino Mega.

**Tabla 7**

*Características del módulo motor vibrador*

| <b>Módulo</b>              | <b>Torque<br/>(kg/cm)</b> | <b>Velocidad<br/>de rotación<br/>(Rpm)</b> | <b>Voltaje<br/>(v)</b> | <b>Corriente<br/>(mA)</b> |
|----------------------------|---------------------------|--|------------------------|---------------------------|
| <b>Motor<br/>vibrador</b>  | ----                      | 9000                                       | 3 - 5                  | 60-90                     |
| <b>Servomotor<br/>SG90</b> | 1.8                       | ---  | 5                      | 150                       |

**Fuente.** Tomada de (Sando, 2022), (Robots, 2022)

Para la selección del servomotor se tuvo en cuenta el brazo del servo que es de 2 centímetros y el grado de inclinación máxima para mover el exoesqueleto es de 60° y se calcula usando la Ecuación 1.

$$Peso\ max = 1.5\ kg * 2\ cm * \sin(60^\circ) = 2.59\ kg * cm$$

Como se observa el peso máximo que se obtuvo con un servo de 1.8 kg\*cm y un ángulo de inclinación máxima de 60° es 3.11 kg\*cm por lo que es suficiente para que el guante despliegue la fuerza necesaria.

Para el guante háptico se tiene 5 servomotores de 1.8 kg/cm los cuales controlan el exoesqueleto para la retroalimentación de fuerza, también un motor vibrador por cada dedo el cual funciona a voltajes de 5 V suficiente para trabajar con Arduino.

## 2.5.5 Giroscopio

Se detalló las características del módulo giroscopio MPU 6050, la Tabla 8 teniendo en cuenta los voltajes de trabajo, la corriente y la sensibilidad (dps) que es la velocidad angular para cada eje en grados por segundo.

**Tabla 8**

*Características del módulo giroscopio Mpu-6050*

| Módulo   | Sensibilidad (dps)           | Voltaje (v) | Corriente (mA) |
|----------|------------------------------|-------------|----------------|
| MPU-6050 | ±250, ±500, ±1000, y ±2000   | 3.3 - 5     | 20             |
| GY-521   | ± 250, ± 500, ± 1000, ± 2000 | 2.6 - 5     | 3.5            |

**Fuente.** Tomada de (Naylamp, 2022), (Electronica, 2022)

Para el giroscopio se optó el módulo MPU-6050, ya que cuenta con 6 ejes los cuales ayudan a la estabilidad para detectar los movimientos del giroscopio, ayudando a detectar la dirección del vehículo, en este caso izquierda, derecha y centro, otro punto a detallar es lo accesible del módulo, ya que se encuentra en varias partes de electrónicas.

## 2.5.6 Capacidad de la batería para el guante háptico

Para el prototipo se utilizó 2 baterías tipo Li-Po, ya que una batería brinda energía al guante háptico mientras que la segunda batería energiza al vehículo, para ello se tiene en cuenta la corriente y el voltaje que consumen todos los componentes que integran este prototipo por lo que se detalló en la Tabla 9 todos los componentes seleccionados.

**Tabla 9**

*Componente para el guante háptico*

| Materiales                 | Cantidad | Voltaje (v) | Corriente (mA) | Corriente Total (mA) |
|----------------------------|----------|-------------|----------------|----------------------|
| Arduino Mega               | 1        | 7 - 12      | 200            | 200                  |
| Sensor Flex                | 6        | 5 - 12      | 20             | 120                  |
| Motor vibrador             | 5        | 3.3 – 5.3   | 90             | 450                  |
| Servomotor SG90            | 5        | 5           | 150            | 750                  |
| Mpu-6050 giroscopio        | 1        | 3.3 – 5     | 20             | 20                   |
| LoRa Shield                | 1        | 3.3 - 5     | 150            | 150                  |
| <b>Corriente Tota (mA)</b> |          |             |                | 1690                 |

**Fuente.** Tomada de (Meier, 2020), (Rodriguez D. , 2018), (Rambal, 2022), (Naylamp, 2022)

Como se observa en la Tabla 9 se presenta la corriente de cada componente y en algunos casos como se utiliza varios componentes del mismo tipo se multiplica el valor de la corriente por el número de elementos, al final realizamos una sumatoria de la corriente total de cada componente y como resultado se obtuvo 1690 mA.

A continuación, se calcula la corriente de la batería necesaria para el funcionamiento del guante háptico. Como se obtuvo una corriente total de 1690 mA que es la sumatoria de todas las corrientes de los componentes electrónicos y se optó para trabajar un tiempo determinado de una 1 hora lo suficiente como para realizar pruebas y ajustes del prototipo, para ello se calcula la corriente de la batería por medio de la Ecuación 2.

$$\text{Corriente}/h = 1690 \frac{\text{mA}}{h}$$

$$\text{Tiempo} = 60 \text{ min} = 1 \text{ h}$$

$$\text{Batería} = \text{Corriente} * \text{Tiempo} \quad \text{Ecuación 2}$$

$$\text{Batería} = 1690 \frac{\text{mA}}{h} * 1 \text{ h} = 1690 \text{ mA}$$

Mediante los cálculos realizados se concluyó que, para un tiempo de uso de una hora, se obtuvo un consumo de corriente de 1690 mA, con esto se optó por elegir para el guante háptico una batería Li-Po de 7.4 voltios con un consumo de corriente de 1.7 A/h que en el mercado lo más aproximado es de 2.2 A/h.

## 2.6 Componentes que integran el vehículo

Para los componentes del vehículo se realizó una comparación entre tecnologías en cuanto a controladores sensores y actuadores con el fin de detallar los componentes que se utilizan para la implementación, al igual que el tipo de batería acorde al uso que se le dio.

### 2.6.1 Controlador para el vehículo

A continuación, se detalló las características de los controladores Arduino, ya que es del conjunto de dispositivos programables, para lo que se tiene aspectos detallados como velocidad de procesamiento precio y rendimiento, por lo que se detalla en la Tabla 10 las siguientes características.

**Tabla 10***Características de la familia de tecnologías de Arduino*

| Arduino      | Entradas y salidas digitales | Entradas y salidas analógicas | Voltaje (v) | Corriente (mA) | Voltaje de operación (v) | Clock Speed (MHz) | Peso (g) |
|--------------|------------------------------|-------------------------------|-------------|----------------|--------------------------|-------------------|----------|
| <b>Due</b>   | 54                           | 12                            | 7 - 12      | 200            | 3.3                      | 84                | 36       |
| <b>Mega</b>  | 54                           | 12                            | 7 - 12      | 200            | 3.3 – 5                  | 16                | 37       |
| <b>Uno</b>   | 14                           | 6                             | 7 - 12      | 60             | 3.3 – 5                  | 16                | 25       |
| <b>Nano</b>  | 22                           | 8                             | 7 - 12      | 19             | 3.3 – 5                  | 16                | 7        |
| <b>Esp32</b> | 16                           | 2                             | 3.3 - 5     | 19             | 3.3                      | 16                | 12       |

**Fuente.** Tomado de (Roda, 2020), (Llamas, 2017), (Meier, 2020)

Con la Tabla 10 de características técnicas todos los controladores Arduino funcionan de 7 a 12 voltios, para lo cual se optó nuevamente por el Arduino Mega, ya que se necesita una correcta comunicación con el módulo Shield LoRa disponiendo entradas analógicas, digitales y cumple con el funcionamiento del vehículo además de tener pines de conexión lo que evita que se realice una placa de expansión adicional para poder utilizar los pines.

### 2.6.2 Motores o actuadores

A continuación, se detalló los motores o actuadores que se ocupan para el vehículo que se va a controlar con algunas características especificadas en la Tabla 11.

**Tabla 11***Características de los actuadores*

| Actuadores                        | Velocidad (Rpm) | Torque (kg/cm) | Voltaje (v) | Corriente (mA) |
|-----------------------------------|-----------------|----------------|-------------|----------------|
| <b>Motorreductor Pololu 100:1</b> | 310             | 1.7            | 5 – 12      | 75             |
| <b>Servomotor MG90S</b>           | ---             | 12             | 4 – 7.2     | 150            |
|                                   |                 | 2.2            | 4.8 – 6.6   | 130            |

**Fuente.** Tomado de (Robots, 2022), (Carrod, 2022)

En la Tabla 11 se observa un actuador motorreductor Pololu 100:1 para la tracción trasera del vehículo y un Servomotor MG90S para la tracción delantera del mismo.

Para la selección del servomotor del vehículo se tiene en cuenta el brazo del servo que es de 2 centímetros y el grado de inclinación máxima para mover la dirección del vehículo es de 90° y se calculó usando la Ecuación 1.

$$\text{Peso max} = 2.2 \text{ kg} * 2 \text{ cm} * \text{sen}(90^\circ) = 4.4 \text{ kg} * \text{cm}$$

Como se observó, el peso máximo que se obtiene con un servo de 2.2 kg\*cm y un ángulo de inclinación máxima de 90° es 4.4 kg\*cm por lo que es suficiente dar el giro necesario para la dirección y soporte de todos los componentes del vehículo.

Para el vehículo que se controló por medio del guante háptico se tiene en cuenta dos motorreductores Pololu 100:1 para comandar la fuerza, velocidad del vehículo, ya que este motorreductor cuenta con una velocidad baja, pero con mayor torque lo suficiente para soportar todos los componentes del vehículo y un servomotor para comandar el giro o dirección con un torque de 2.2 kg/cm.

### 2.6.3 Sensores para el vehículo

Se realizó un análisis entre algunos tipos de sensores de los cuales tenemos sensores fotoeléctricos y ultrasónicos, los cuales se detallan las características en la Tabla 12.

**Tabla 12**

*Características de sensores*

| <b>Tipos de Sensores</b>    | <b>Clases</b> | <b>Voltaje (v)</b> | <b>Corriente (mA)</b> | <b>Distancia (cm)</b> |
|-----------------------------|---------------|--------------------|-----------------------|-----------------------|
| <b>Fotoeléctricos Sharp</b> | GP2Y0A41SK0F  | 4.5 - 5.5          | 12                    | 4 - 30                |
|                             | GP2Y0A21YKF0  | 4.5 - 5.5          | 12                    | 10 - 80               |
|                             | GP2Y0A02YK0F  | 4.5 - 5.5          | 12                    | 20 - 150              |

**Fuente.** Tomada de (Leibson, 2018)

Para el vehículo se optó por los sensores Sharp de distancia GP2Y0A21YKF0 de 10 a 80 centímetros, ya que estos sensores la respuesta de medición es más rápida y precisa a diferencia de los sensores ultrasónicos que su respuesta es retardada. Estos sensores Sharp se usan para censar objetos y envíe, una señal al guante háptico para indicar que el vehículo está próximo a un obstáculo.

## 2.6.4 Capacidad de la batería para el vehículo

Para el cálculo de la segunda batería para el vehículo se tiene en cuenta la corriente y el voltaje que consumen todos los componentes que integran este prototipo, por lo que se detalló en la Tabla 13 todos los componentes a utilizados.

**Tabla 13**

*Componentes para el vehículo*

| Materiales                            | Cantidad | Voltaje (v) | Corriente (mA) | Corriente Total (mA) |
|---------------------------------------|----------|-------------|----------------|----------------------|
| Arduino Mega                          | 1        | 7 - 12      | 200            | 200                  |
| Sensor Sharp 10 - 80cm                | 4        | 4.5 - 5     | 12             | 96                   |
| Motorreductores Pololu 100:1 y driver | 2        | 5 - 12      | 75             | 150                  |
| Servomotor 2.2 kg/cm                  | 1        | 4.8 – 6.6   | 130            | 130                  |
| LoRa Shield                           | 1        | 3.3 - 5     | 150            | 150                  |
| <b>Corriente Total (mA)</b>           |          |             |                | <b>726</b>           |

**Fuente.** Tomada de (Leibson, 2018), (Roda, 2020), (Llamas, 2017), (Meier, 2020), (Robots, 2022), (Carrod, 2022)

Una vez calculado la batería número 1, se procede a calcular la corriente de la batería número 2 necesaria para el funcionamiento del vehículo. Cómo se obtuvo una corriente total de 726 mA y se optó para trabajar un tiempo determinado de una hora lo suficiente como para realizar pruebas y ajustes del prototipo, con esto se calcula la corriente de la batería por medio de la Ecuación 2.

$$\text{Corriente}/h = 726 \frac{\text{mA}}{h}$$

$$\text{Tiempo} = 60 \text{ min} = 1 \text{ h}$$

$$\text{Batería} = 726 \frac{\text{mA}}{h} * 1 \text{ h} = 726 \text{ mA}$$

Mediante los cálculos realizados se concluyó que para un tiempo de uso de una hora se obtuvo como resultado un consumo de corriente de 726 mA para el vehículo, por lo que es necesario una batería Li-Po de 7.4 voltios con un consumo de corriente de 726 mA/h pero el valor comercial encontrado es de 1100 mA/h.

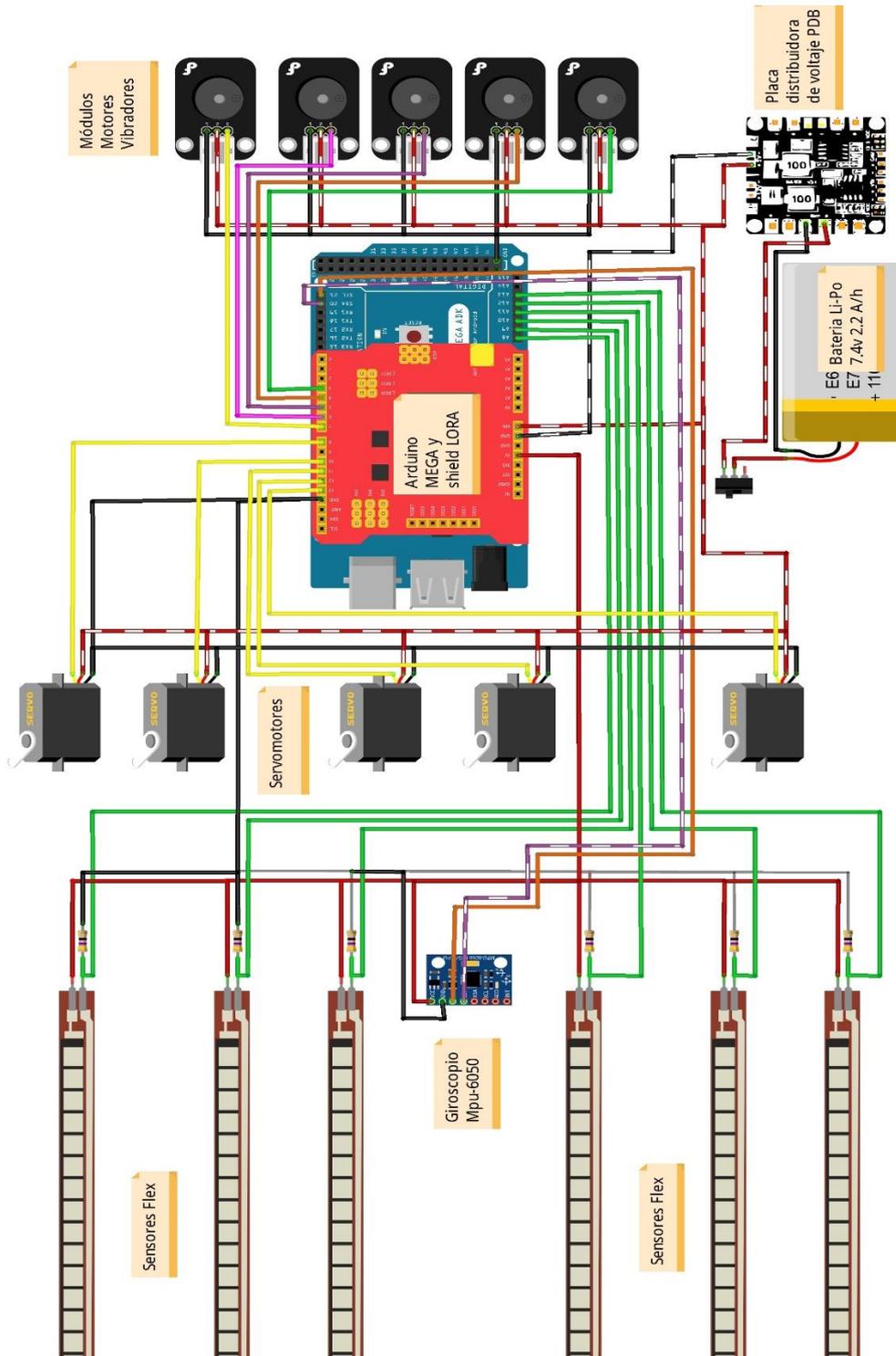
## **2.7 Diseño electrónico del guante háptico**

En base en la selección de materiales se realizó el diseño electrónico en el cual se detalla todos los componentes con sus respectivas conexiones tanto como para el guante háptico como para el vehículo, con el fin de detallar la forma que tendrá el diseño electrónico.

Una vez seleccionado los materiales que integraran el guante háptico, se organizó un diagrama de las conexiones de todo el diseño eléctrico, en donde los pines que se utilizan tanto como entradas analógicas y digitales se observa en la Figura 29.

**Figura. 29**

*Diagrama electrónico del guante háptico*



**Fuente. Autor**

La batería Li-Po de 7.4 voltios es la que alimenta a todo el circuito electrónico que controla los componentes, el guante háptico, para ello el Arduino Mega, los servomotores, los módulos motor-vibradores una placa distribuidora de voltaje PDB con el fin de reducir el ruido que se genera en el Arduino y proteger a los servomotores de sobrevoltajes de la batería. Los sensores Flex, y el giroscopio MPU 6050 se conecta a la alimentación de 5 voltios que ofrece el Arduino debido a que es el voltaje que soportan estos componentes.

Para la conexión del sensor Flex se conecta un pin a los 5 voltios del Arduino, mientras que el segundo pin se conecta a una entrada analógica y a una resistencia de 20 Kiloohmios, y del extremo de la resistencia se conecta a GND, este sensor se colocó en los dedos de un guante elástico para facilitar su flexibilidad.

El módulo motor vibrador consta de 3 pines, los cuales son VCC, GND, SEÑAL, para lo cual se alimenta con 5v que ofrece la placa distribuidora de voltaje, GND, y el pin de SEÑAL se conecta a una entrada digital del Arduino desde el pin 7 al 3.

El servomotor consta de 3 pines los cuales son VCC, GND, SEÑAL, para lo cual se alimenta con 5v que ofrece la placa distribuidora de voltaje, GND, y el pin de SEÑAL se conecta a una entrada digital del Arduino desde el pin 13 al 8.

Para el giroscopio Mpu 6050 usa una comunicación I2C, consta de 8 pines de los cuales se ocuparán 4 pines que son VCC, GND, SCL y SDA que son los pines del giroscopio, ya que tiene el resto de los pines para el velocímetro este dispositivo se alimenta con 5 voltios su respectivo GND, el pin SDA se conecta a una entrada SDA del Arduino que es el pin 20, al igual que él SCL que es el pin 21.

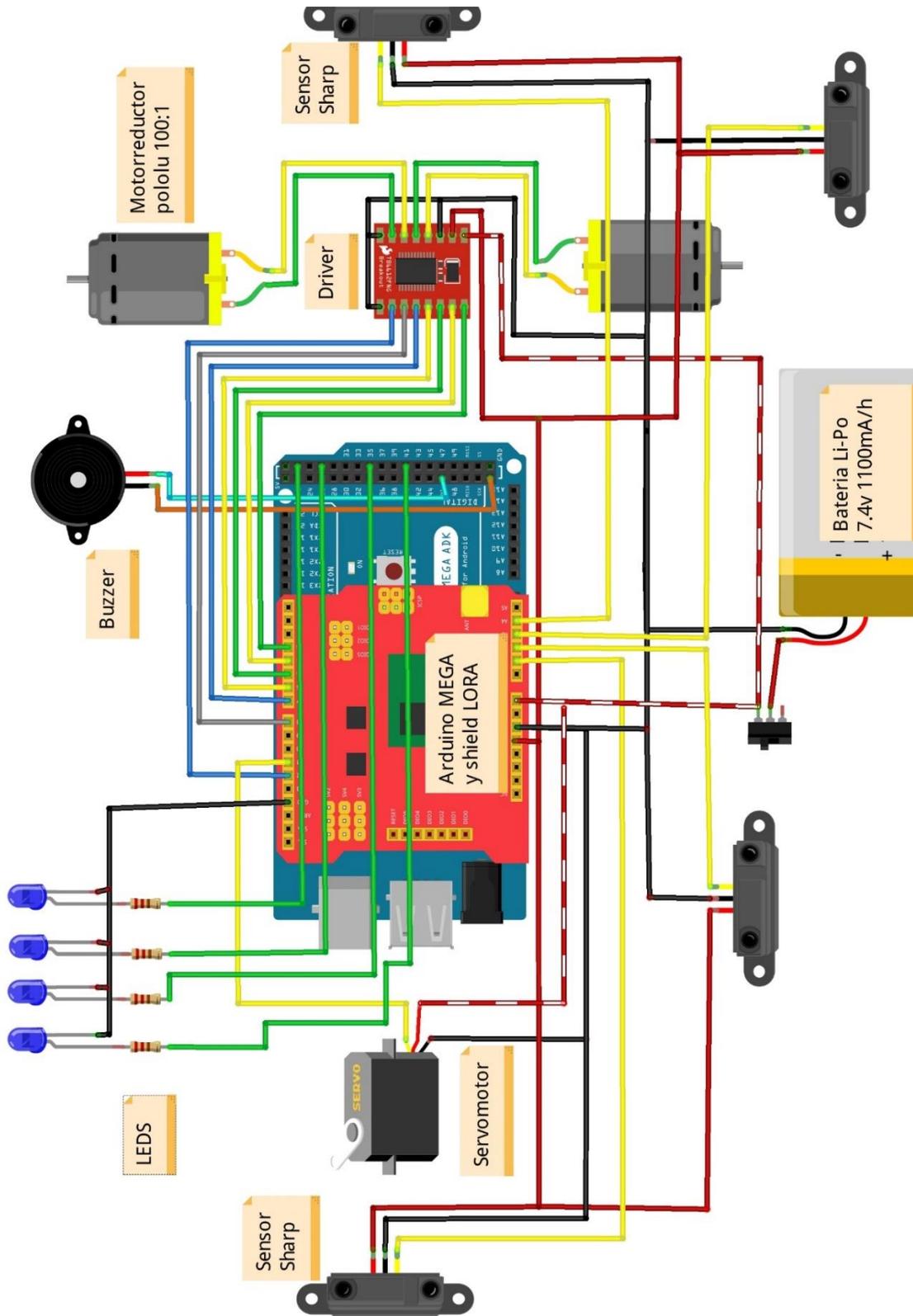
La Shield LoRa se energiza por medio del Arduino Mega por lo tanto se acopla en todos los pines para facilitar el cableado y mejorar la comunicación de una forma más eficiente, ya que cuenta con una antena para mejorar el alcance de comunicación.

## **2.8 Diseño electrónico del vehículo**

Luego de seleccionar los materiales que intervienen en el armado del vehículo, se organizó un diagrama de las conexiones de todo el diseño eléctrico, en donde los pines que se utilizan tanto como entradas analógicas y digitales se observan en la Figura 30.

**Figura. 30**

*Diagrama electrónico del vehículo*



Fuente. Autor

La batería Li-Po de 7.4 voltios es la que alimenta a todo el circuito electrónico que controla los componentes del vehículo, para ello el Arduino Mega, el servomotor, los motores Pololu con su driver se conecta a la batería el fin de disminuir el ruido que se genera en el Arduino por lo que los 4 sensores Sharp de distancia y Buzzer, se conecta a la alimentación de 5 voltios que ofrece el Arduino debido a que es el voltaje que soportan estos componentes, todos los componentes mencionados se colocan en la base del vehículo como método de prueba.

Se colocan 4 leds indicadores, los cuales interactúan con el guante flexionado de cada dedo que sirve como direccional delantero para la izquierda, derecha, y direccional trasero izquierda, derecha.

Los sensores Sharp constan de 3 pines los cuales son VCC, GND, SEÑAL, para ello se conectan a los 5 voltios que genera el Arduino Mega, su respectivo GND y la señal se conecta a una entrada analógica del Arduino que están distribuidos desde el A1 al A4.

El servomotor consta de 3 pines los cuales son VCC, GND, SEÑAL, para ello se conectan a los 7.4 voltios que genera la batería, su respectivo GND y la señal se conecta a una entrada digital del Arduino Mega que tenga PWM para poder controlar el ángulo de giro que es hasta 180°.

El driver se alimenta por la misma fuente de energía de la batería, en las salidas OUT1 y OUT2 se conecta el primer motorreductor Pololu, en las salidas OUT3 y OUT4 se conecta el segundo motorreductor Pololu, las entradas IN1, IN2, IN3, IN4 se conecta a las entradas digitales del Arduino que tengan PWM con el fin de poder controlar la velocidad de los motorreductores.

El Buzzer es controlado por un sensor Flex que se encuentra ubicado en la palma del guante elástico, el cual da la indicación para que actúe como pito para el vehículo este dispositivo electrónico tiene dos pines que son GND y Señal la cual va conectada al pin 46 del Arduino.

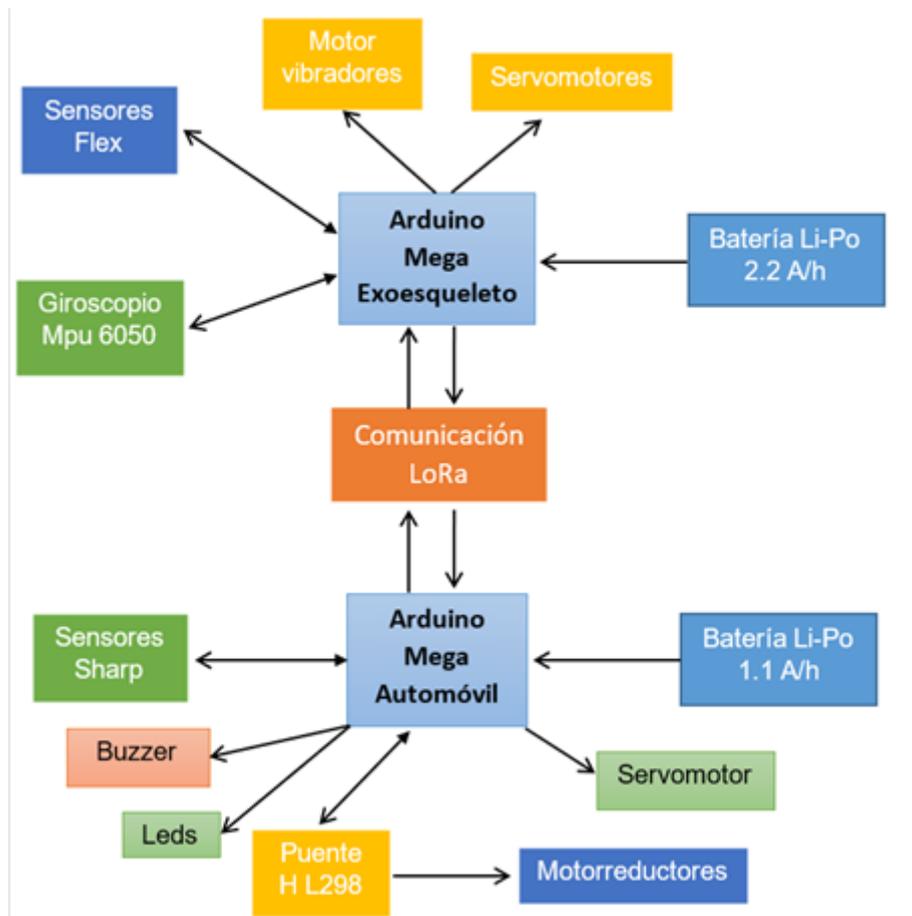
La Shield LoRa se energiza por medio del Arduino Mega, ya que se acopla en todos los pines para facilitar el cableado y mejorar la comunicación de una forma más eficiente y directa, ya que cuenta con una antena para mejorar el alcance de comunicación.

## 2.9 Diagrama de bloques

Para entender cómo funciona del guante háptico se indicó en un diagrama de bloques como se comunica cada componente electrónico en conjunto para ello se presenta la siguiente Figura 31.

**Figura. 31**

*Diagrama de bloques*



**Fuente.** Autor

En la figura 31 se presentan los principales componentes del guante háptico y el vehículo, en el cual para cada componente se tiene sus baterías personales que energizarán a las dos placas de Arduino Mega.

La primera placa de Arduino Mega controla el guante háptico que tiene conectado los Sensores Flex, el giroscopio y los motores vibradores. El Sensor Flex se encarga de enviar una señal al Arduino para poder controlar una de las acciones que tiene el vehículo, de la misma forma el giroscopio enviará una señal al Arduino para que el vehículo realice una

nueva acción y por último el motor vibrador se utiliza cómo una alarma indicativa la cual da a conocer que el vehículo está a punto de estrellarse con un obstáculo u objeto. Los servos realizan la retroalimentación de fuerza del exoesqueleto por lo que hay 5 servos para cada dedo.

La segunda placa de Arduino Mega controla todos los dispositivos electrónicos del vehículo cómo son los sensores Sharp, los actuadores entre ellos el motorreductor Pololu, Buzzer y el servomotor. Por esta razón tanto el guante háptico como el vehículo tienen una comunicación inalámbrica por medio de radiofrecuencia que se usa el módulo LoRa, la cual es bidireccional por lo que ambas partes podrán comunicarse entre sí para enviar y recibir información para poder ejecutar todas las acciones.

## **2.10 Software**

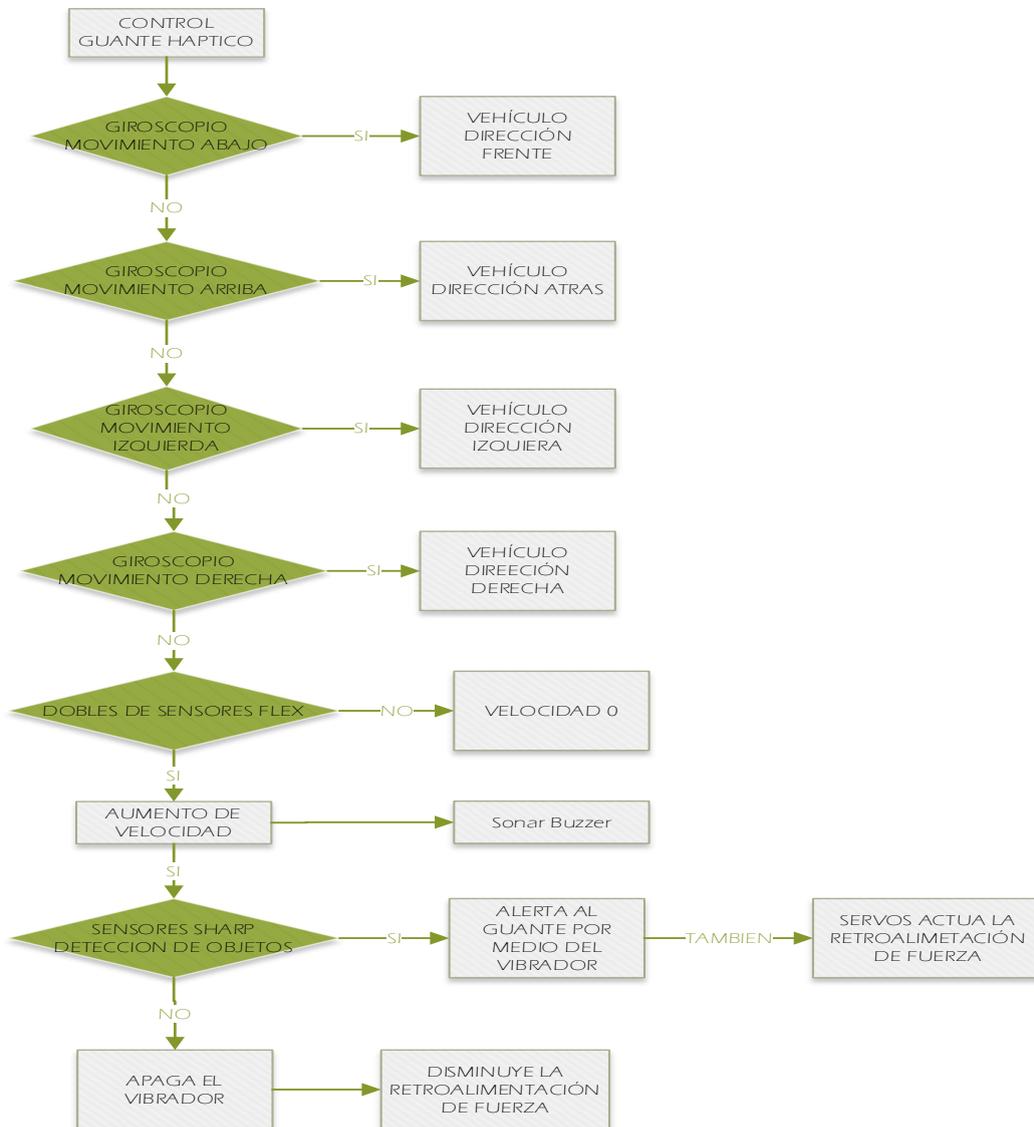
Para la programación se utilizó el software Arduino en el que se realizó el código para los sensores fotoeléctricos de marca Sharp, la interacción y comunicación con el módulo LoRa, la comunicación con el giroscopio, los actuadores, que son motores, los sensores Flex, y el vibrador para lo cual se realiza una programación para cada dispositivo en este caso una que contendrá el Arduino Mega que será el encargado de controlar el guante háptico, y otra programación contendrá el Arduino Mega para el control del vehículo.

La comunicación la realizan los dispositivos LoRa ubicados en el vehículo y el guante, por lo que es una comunicación bidireccional, puesto que entre ambos se envían información para comandar las acciones programadas.

## **2.11 Flujograma**

Para indicar el funcionamiento del prototipo se presenta un flujograma, el mismo que indica las condiciones que tendrá tanto el guante háptico como el vehículo, ya que ambos se comunican entre sí para efectuar la lógica programable, como se muestra en la Figura 32.

**Figura. 32**  
*Flujograma*



**Fuente.** Autor

Una vez realizado el análisis se establece los parámetros y condiciones que requiere el guante háptico con retroalimentación de fuerza, cada dispositivo electrónico que conforma el guante es empleado a través de la calidad de respuesta y el cumplimiento de características solicitadas obteniendo así el diseño de control, comunicación y mecánico del prototipo apoyado en un modelo de exoesqueleto existente, añadiendo pequeñas modificaciones propias para la aplicación del mismo, donde es un factor importante para la obtención de los parámetros y resultados deseados conjuntamente con el desarrollo correcto del funcionamiento.

## **CAPÍTULO 3**

### **Implementación del guante háptico con retroalimentación de fuerza**

En el presente capítulo se muestra el ensamblaje físico y la implementación de los componentes estudiados anteriormente, conjuntamente con el funcionamiento, análisis de fallas mecánicas y electrónicas, pruebas, resultados con respecto a la implementación de un guante háptico con retroalimentación de fuerza controlado a través de un exoesqueleto para la teleoperación de un vehículo no tripulado.

#### **3.1 Impresión 3D del guante háptico con retroalimentación de fuerza**

El exoesqueleto fue tomado de un modelo existente (Curtis, 2021), que posteriormente fue modificado parcialmente, utilizando el software de diseño 3D Solidworks para el diseño e impresión de las partes y cumplir con los parámetros establecidos.

El diseño está desarrollado para su utilización en la mano derecha de un adulto en caso de que se desee utilizar en una mano más pequeña como la de un niño debemos de reducir la escala e imprimir nuevamente el diseño, también si deseamos usarlo en la mano izquierda debemos de girarlo totalmente en modo espejo en el programa e imprimirlo.

Para el proceso de impresión de las distintas partes que conforman el exoesqueleto del guante háptico se tomó en cuenta las distintas calidades y marcas existentes en cuanto a materiales de impresión 3D, se optó por utilizar la fibra de tereftalato de polietileno Glicol conocida con las siglas (PETG) mostrado en la Figura 33, debido a sus propiedades como la alta resistividad al impacto y es ideal para ser sometido a una tensión mecánica. (TRACTUS3D, 2022).

### Figura. 33

*Fibra para impresión 3D- PETG*

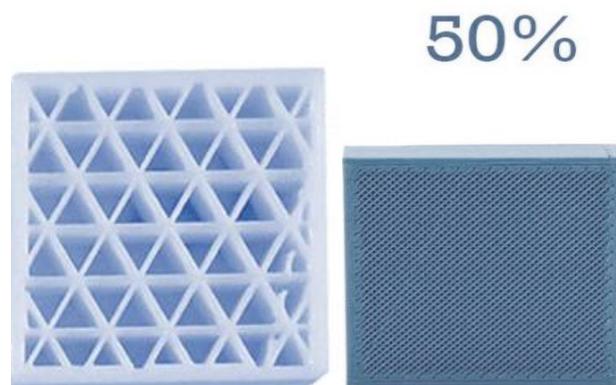


**Fuente.** Tomado de (SUNLU, 2022)

En la ejecución de la impresión del guante háptico se utilizó un relleno del 50% para obtener una mayor firmeza en el cruce de las líneas de impresión y una mayor densidad de la pieza, con un patrón de relleno triangular como se muestra en la Figura 34. Se utilizó una temperatura de la cama de impresión 3D entre 75°C a 80°C y la temperatura del bloque calefactor 230°C a 260°C, siendo estos valores necesarios para la correcta impresión de la fibra de PETG.

### Figura. 34

*Tipos de patrón de relleno en impresión 3D*



**Fuente.** Tomado de (Filament2Print, 2022)

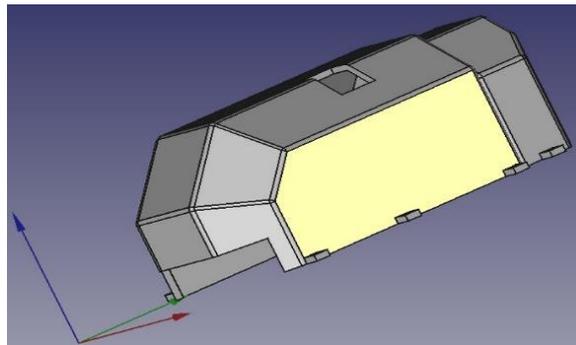
Los tiempos de impresión de cada una de las piezas fueron variables, ya que dependieron de la velocidad y de la cantidad de relleno que se aplicó, también al momento de realizar las pruebas hubo en defecto en el pulgar del exoesqueleto por lo que se tuvo que volver a imprimir

dicha pieza, dando como resultado un total de 51 horas de impresión en todo el guante háptico.

Se realizó el diseño e impresión de una pieza para cubrir todo el sistema de control del vehículo, para ello se eligió el filamento ácido poliláctico conocido por las siglas (PLA) que según (SICNOVA, 2022) es un material con un grado de dureza y deformidad muy baja siendo el más idóneo para la impresión de la pieza, todo esto a una temperatura de 230°C en extrusor y 55°C en base calefactada con un patrón de relleno octeto, este tuvo una duración de 18 horas de impresión como se muestra en la Figura 35.

**Figura. 35**

*Carcasa del sistema de control del vehículo*



**Fuente.** Autor

### **3.2 Ensamblado del exoesqueleto**

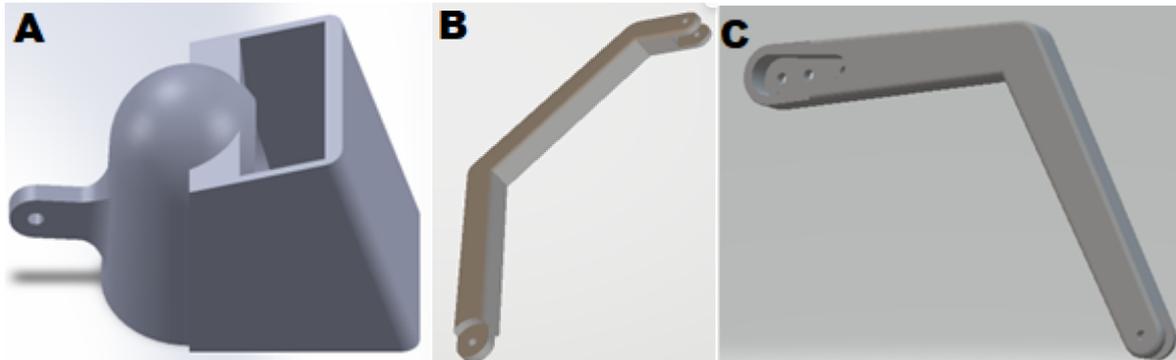
El exoesqueleto fue ensamblado con tornillos de 3mm y tuercas milimétricas para la sujeción de los servomotores a la base del exoesqueleto, se utilizó también los tornillos que proporciona el fabricante del servomotor para el acople de las hélices con las piezas de los dedos y pasadores metálicos para el armado de cada uno de ellos.

La parte mecánica del exoesqueleto está formada por 2 secciones, dedos y el dorso de la mano, la primera sección está conformada por las piezas que integran cada uno de los 5 dedos de la mano derecha de la persona, donde interviene el asentamiento de cada uno de los 5 motores vibradores en la punta de cada dedo como se muestra en el literal A, por otro lado, un brazo que conecta a la hélice del servomotor como se muestra en el literal C que sustituye a los metacarpianos de la mano humana, finalmente un brazo de unión de las dos piezas como se muestra en el literal B que asemeja a las falanges proximal, medial, distal de

la mano en la Figura 36. Todo esto para obtener una mejor movilidad y respuesta en los dedos.

**Figura. 36**

*Parte 1 de los dedos del exoesqueleto*

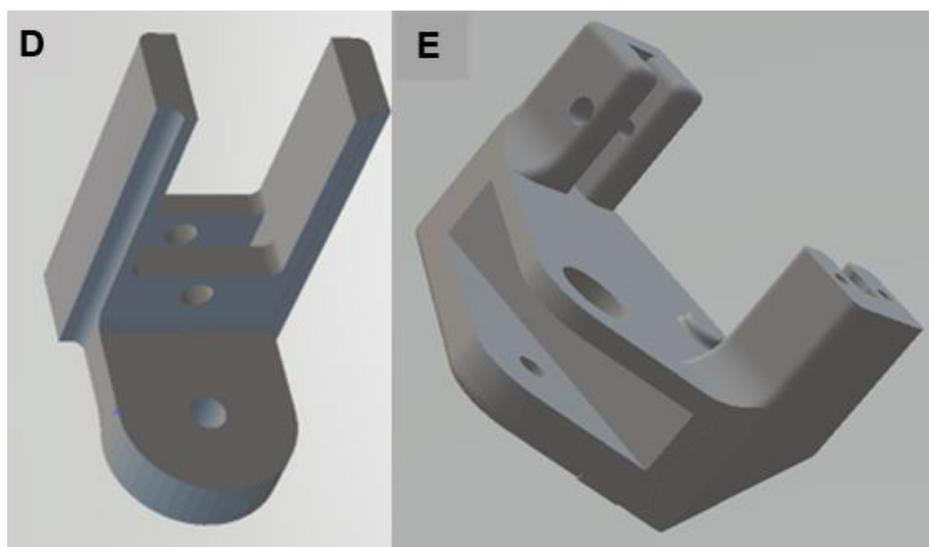


**Fuente.** Autor

Para la segunda sección interviene el asentamiento de los servomotores por medio del cuerpo y la hélice del mismo a la base del dorso de la mano derecha como se muestra en el literal D de la Figura 37, cabe recalcar que para el dedo pulgar se realizó otra pieza completamente distinta, ya que tiene diferente movilidad con respecto a los otros 4 dedos de la mano, pero con los mismos principios de sujeción como se muestra en el literal E de la Figura 37.

**Figura. 37**

*Parte 2 sujeción de los servomotores del exoesqueleto*

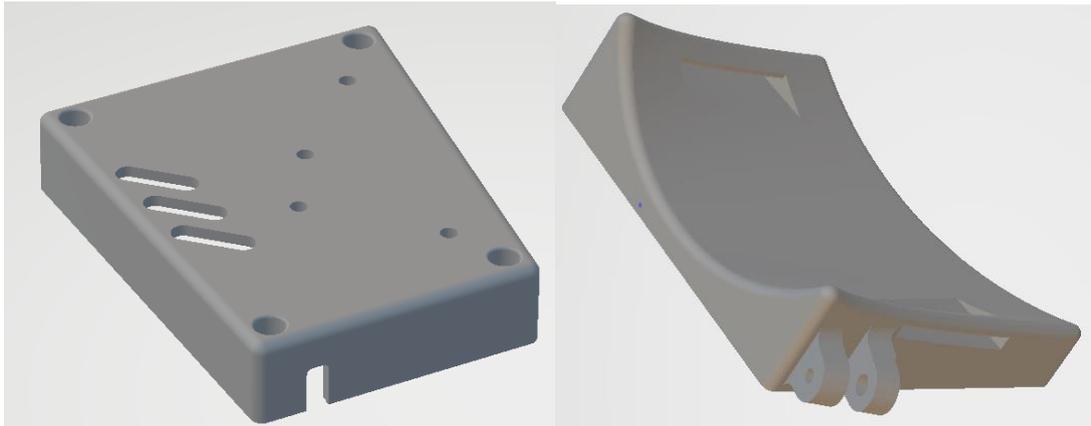


**Fuente.** Autor

Para la tercera sección interviene la unión de las 2 partes antes mencionadas por medio de un compartimiento de sujeción en el dorso de la mano como se muestra en la Figura 38 conjuntamente con un adhesivo velcro para mayor ajuste a la mano del usuario.

**Figura. 38**

*Parte 3 compartimiento del exoesqueleto*

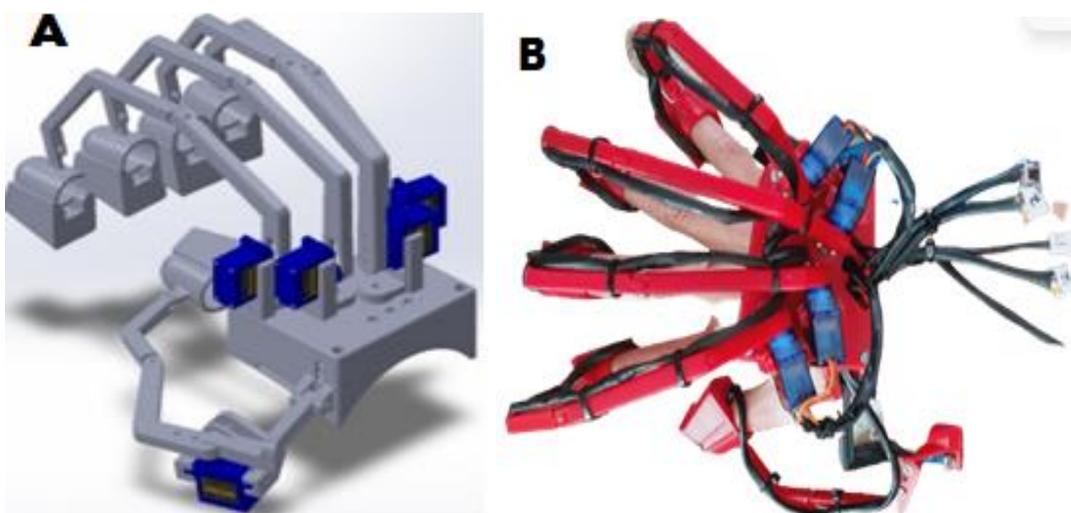


**Fuente.** Autor

Como resultado final se obtiene el exoesqueleto ensamblado con las tres partes ya antes mencionadas como se observa en el literal A de la Figura 39. El literal B nos muestra la fotografía real del exoesqueleto.

**Figura. 39**

*Ensamblaje del exoesqueleto*

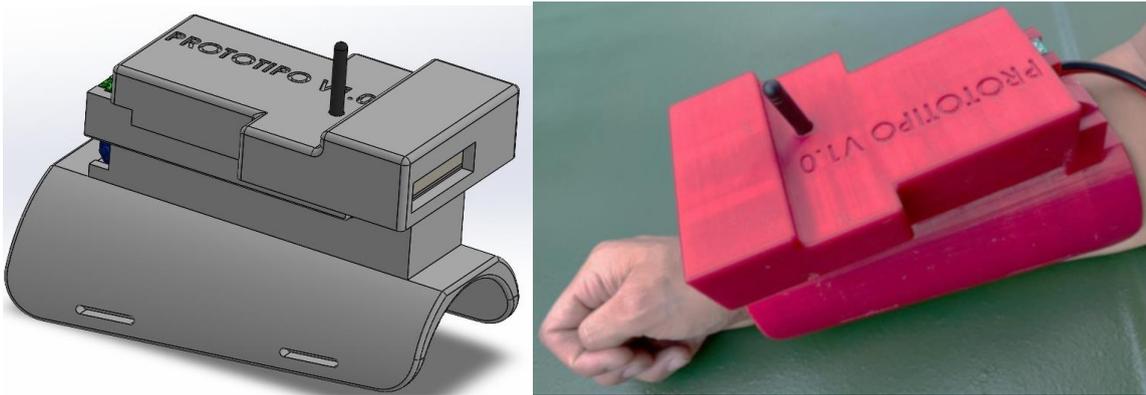


**Fuente.** Autor

Para la parte del montaje de los componentes electrónicos que controlan al exoesqueleto se realizó un compartimiento montado en el antebrazo derecho donde protege y aloja cada uno de los dispositivos electrónicos como se muestra en la Figura 40.

**Figura. 40**

*Compartimiento de componentes electrónicos del exoesqueleto*



**Fuente.** Autor

### **3.3 Distribución y montaje de los componentes electrónicos**

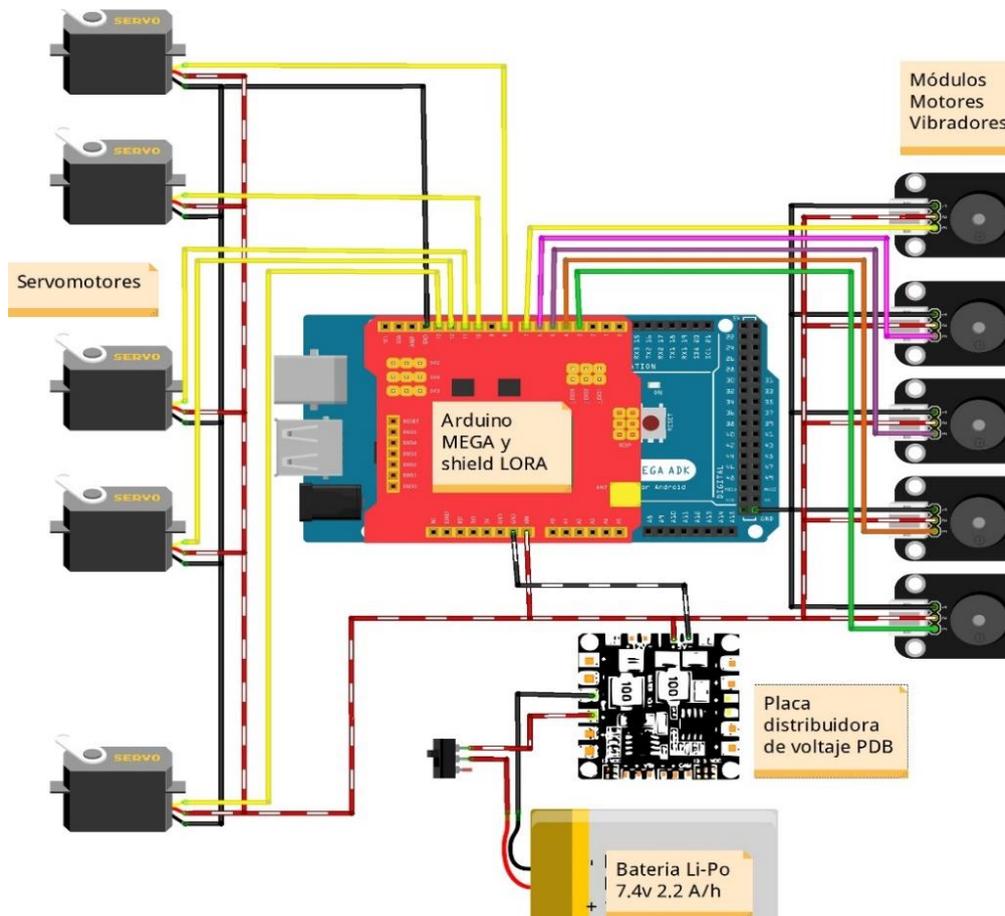
Para el montaje de los dispositivos electrónicos se divide en 2 partes, la primera parte conforma el guante háptico del exoesqueleto, en los cuales consta de un controlador, servomotores, sensores Flex, módulo motores vibradores, giroscopio, Shield LoRa y la batería. La segunda parte intervienen los elementos que conforman el vehículo en los cuales constan de un controlador, un servomotor, motores reductores Pololu y su respectivo driver, sensores Sharp de distancia, Shield LoRa y una batería.

#### **3.3.1 Montaje de los componentes del guante háptico en el exoesqueleto**

Para la colocación de los componentes electrónicos del guante háptico se realizó el montaje de los módulos motores vibradores y servomotores en cada dedo acorde al diagrama de la Figura 41 cabe recalcar que la batería alimenta tanto al Arduino Mega como a los servomotores y motor-vibradores.

**Figura. 41**

*Conexiones de los servomotores y motor vibradores*

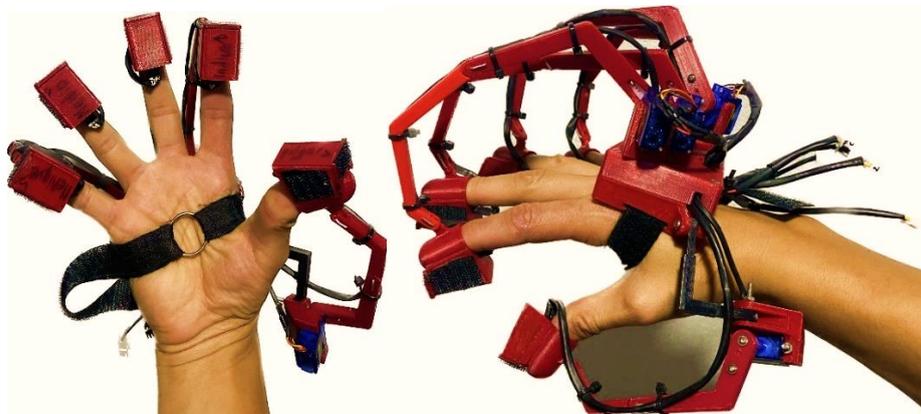


**Fuente. Autor**

A partir de la Figura 41 se ensambló y conectó los componentes en el exoesqueleto impreso en 3D como se observa en la Figura 42.

**Figura. 42**

*Montaje de los servomotores y motor vibradores*

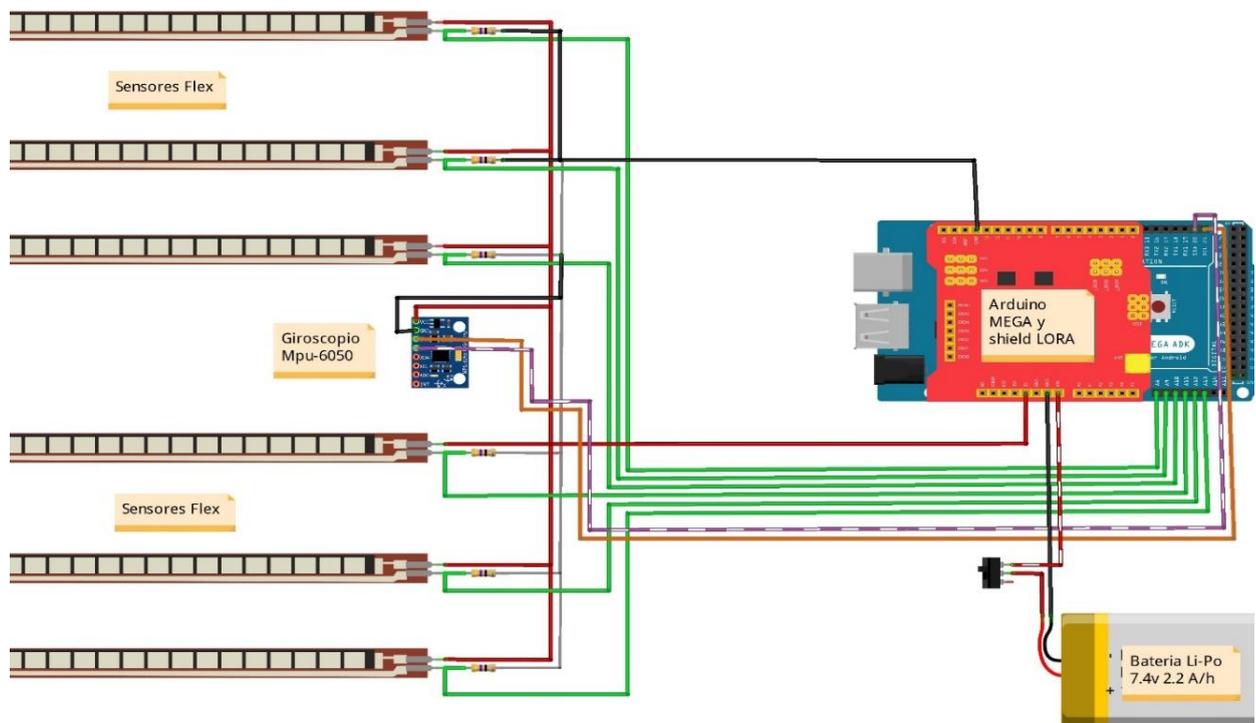


**Fuente. Autor**

A continuación, se procedió a realizar las conexiones de los sensores Flex y el giroscopio para la colocación de los sensores Flex, se realizó el montaje en un guante elástico para poder tener mejor respuesta de los sensores, para ello se siguió el siguiente diagrama que se muestra en la Figura 43.

**Figura. 43**

*Conexiones de los sensores Flex y giroscopio*



**Fuente.** Autor

A partir de la Figura 43 se ensambló y conectó los sensores Flex en el guante elástico que contiene en cada uno de los dedos y la palma un compartimento transparente que facilita la sujeción de cada sensor Flex como se observa en la Figura 44.

**Figura. 44**

*Montaje de los sensores Flex y giroscopio*

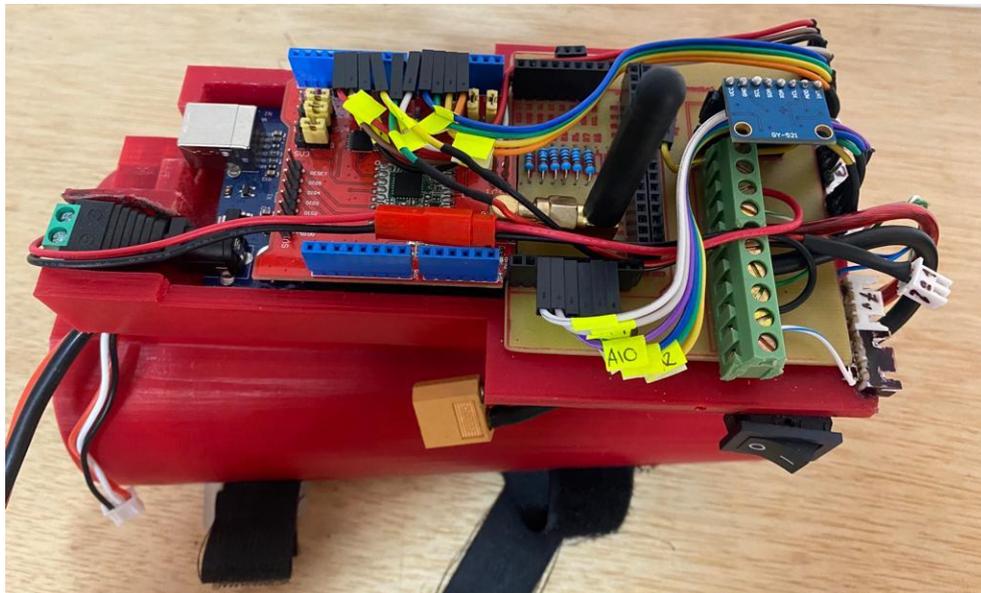


**Fuente.** Autor

Una vez armadas las partes mecánicas de la mano se procedió a ensamblar el interior del antebrazo como se muestra en la Figura 45 con los componentes electrónicos, controlador, módulos de comunicación, Shields expansora de salidas de voltaje y LoRa junto con pines de conexión para el acople de las partes electrónicas del guante que fueron mencionadas anteriormente, todo esto etiquetado para una mayor comprensión de la conexión.

**Figura. 45**

*Montaje de controladores en el compartimiento del antebrazo*

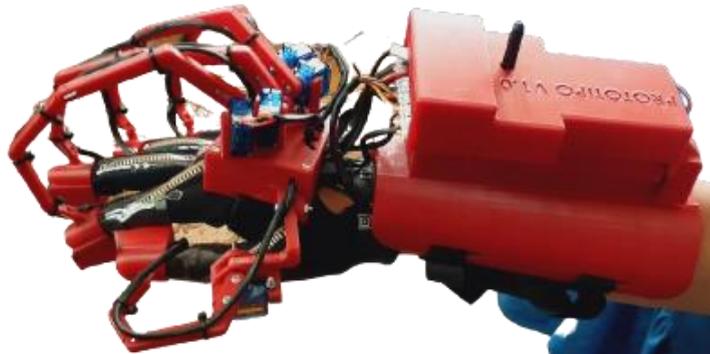


**Fuente.** Autor

Como resultado final se obtuvo el montaje correcto de todos los elementos electrónicos del guante háptico como se muestra en la Figura 46, cabe recalcar que el módulo de comunicación LoRa es una Shield que se acopla al Arduino Mega como se observa en la Figura 45.

**Figura. 46**

*Ensamblaje final del guante háptico*



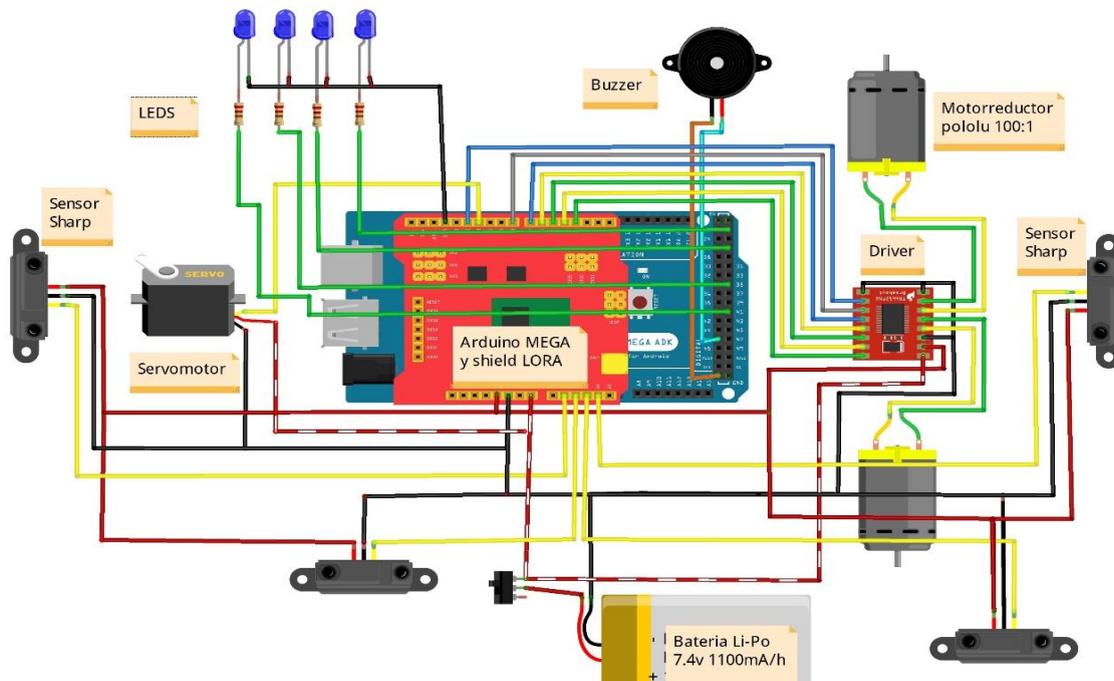
**Fuente.** Autor

### **3.3.2 Montaje de los componentes del vehículo**

Para la colocación de los componentes electrónicos del vehículo se realizó el montaje de los motores Pololu con su respectivo driver y un servomotor para la dirección de giro acorde al diagrama de la Figura 47, cabe recalcar que la batería alimenta tanto al Arduino Mega como a al servomotor y motorreductores Pololu.

**Figura. 47**

*Conexiones del servomotor y motorreductores Pololu*



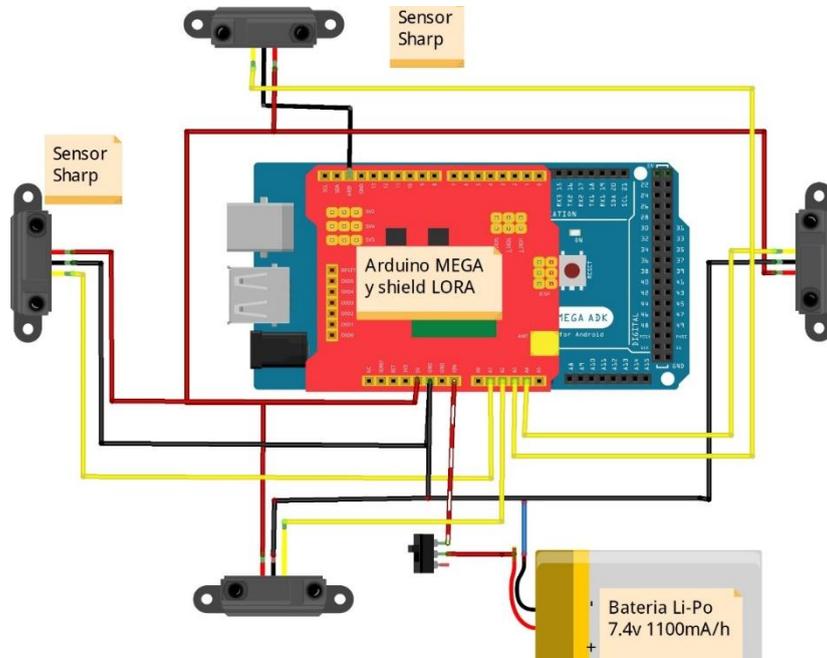
**Fuente.** Autor

Los leds se ubicaron en puntos estratégicos del vehículo, los cuales fueron la parte delantera izquierda, derecha y en la parte trasera izquierda y derecha.

A continuación, se procedió a realizar las conexiones de los cuatro sensores Sharp en puntos específicos para que el vehículo detecte los obstáculos, para ello se siguió el siguiente diagrama que se muestra en la Figura 48.

**Figura. 48**

*Conexiones de los sensores Sharp de distancia*

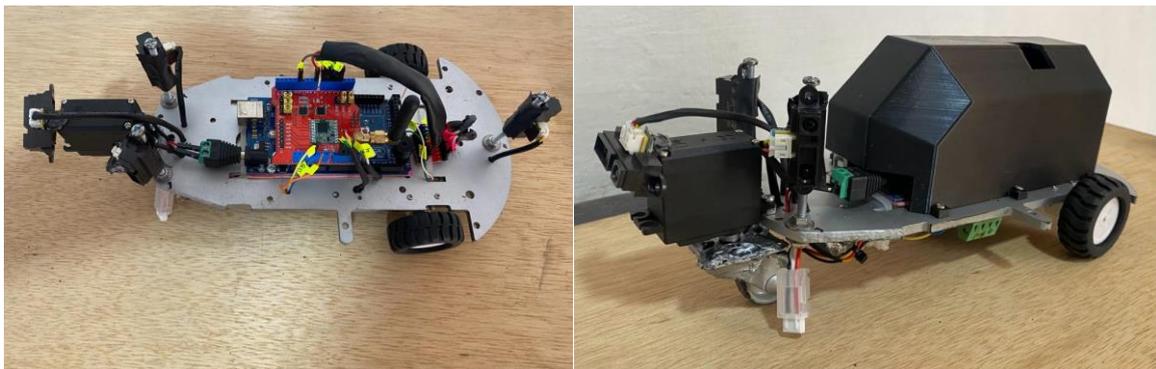


**Fuente.** Autor

Como resultado final se obtuvo el montaje correcto de todos los elementos electrónicos del vehículo, cabe recalcar que el módulo de comunicación LoRa es una Shield que se acopla al Arduino Mega como se observa en la Figura 49.

**Figura. 49**

*Montaje del vehículo finalizado*



**Fuente.** Autor

### 3.4 Desarrollo de Software

Una vez realizado las conexiones electrónicas de ambos dispositivos se procedió a realizar la programación en el software de Arduino IDE, esta plataforma está basada en el lenguaje de programación de alto nivel Processing y está programado en su propio lenguaje, lo que significa que es similar a C++ para el manejo del Hardware. El código de programación de este proyecto se encuentra adjunto en el manual técnico acorde al diseño realizado en el flujograma del Capítulo 2 para el cumplimiento de acciones del prototipo bidireccionalmente.

### 3.5 Trama de datos

A continuación, en la Tabla 14 se muestra los datos enviados y obtenidos del conjunto de sensores y actuadores que conforman el guante háptico con retroalimentación de fuerza

**Tabla 14**

*Trama de datos*

| COMPONENTES                  | TRAMA DE DATOS  |
|------------------------------|---|
| <b>SENSOR FLEX 1,2,3,4,5</b> | El promedio del valor medido de los 5 sensores Flex que oscila entre 200 y 450, se mapea para que genere un valor de 0 a 255, el cual se envía al vehículo para controlar la velocidad.   |
| <b>GIROSCOPIO</b>            | Genera 2 valores, son llamados "X" , "Y", los cuales se envían al vehículo para controlar la dirección izquierda y derecha con el dato de la variable X, mientras que el valor Y controla la marcha y reversa del vehículo, estas dos variables oscilan entre -50 a 50. |
| <b>SENSOR FLEX 1</b>         | Envía señal para accionar direccional delantero derecho del vehículo, los valores oscilan entre 400 a 450.  |

|                               |  |
|-------------------------------|--|
| <b>SENSOR FLEX 2</b>          | Envía señal para accionar direccional delantero izquierdo del vehículo, los valores oscilan entre 400 a 450.   |
| <b>SENSOR FLEX 3</b>          | Envía señal para accionar direccional trasero derecho del vehículo, los valores oscilan entre 400 a 450.   |
| <b>SENSOR FLEX 4</b>          | Envía señal para accionar direccional trasero izquierdo del vehículo, los valores oscilan entre 400 a 450.   |
| <b>SENSOR FLEX 5</b>          | Envía señal para accionar luces de parque de todo el vehículo, los valores oscilan entre 400 a 450.  |
| <b>SENSOR FLEX 6</b>          | Envía señal para accionar bocina del vehículo, los valores oscilan entre 400 a 450.  |
| <b>SENSOR SHARP DERECHO</b>   | Envía señal al guante para accionar el motor vibrador 1 cada vez que se detecte un obstáculo a una distancia de 15 cm, además envía una señal para accionar a los servomotores para que el exoesqueleto se abra. |
| <b>SENSOR SHARP DELANTERO</b> | Envía señal al guante para accionar el motor vibrador 2 cada vez que se detecte un obstáculo a una distancia de 15 cm, además envía una señal para accionar a los servomotores para que el exoesqueleto se abra. |
| <b>SENSOR SHARP IZQUIERDO</b> | Envía señal al guante para accionar el motor vibrador 3 cada vez que se detecte un obstáculo a una distancia de 15 cm, además envía una señal para accionar a los servomotores para que el exoesqueleto se abra. |
| <b>SENSOR SHARP TRASERO</b>   | Envía señal al guante para accionar el motor vibrador 4 cada vez que se detecte un obstáculo a una distancia de 15 cm,   |

---

además envía una señal para accionar a los servomotores para que el exoesqueleto se abra.

---

**Fuente.** Autor

### **3.6 Pruebas**

Para corroborar el funcionamiento del guante háptico con retroalimentación de fuerza se realizaron distintas pruebas donde están involucradas, la duración de las baterías, rapidez y fluidez de la comunicación de ambos dispositivos, respuestas vibratorias, retroalimentación de fuerza con respecto a la comunicación del vehículo, comunicación del giroscopio, respuesta de sensores y actuadores. Las pruebas se realizaron manteniendo un movimiento y uso continuo en una cancha deportiva de futbol sala que fue medida por 40 metros de largo por 20 metros de ancho, como se muestra en la Figura 50.

**Figura. 50**

*Cancha de Pruebas*



**Fuente.** Autor

#### **a) Carga y duración de baterías**

Antes de realizar las pruebas se procedió a cargar las baterías. En esta sección se midió acorde el tiempo establecido en los parámetros, para ello se realizó distintas

pruebas en las cuales ambas baterías se cargaron al 100% con un voltaje de 8.2 V en la batería del vehículo y 8.4 V en la batería del guante, posteriormente se verificó que cumplen con la hora de uso con respecto a los valores calculados en el capítulo 2, obteniendo un voltaje de 4.2 V en la batería del vehículo y 4.3 V en la batería del guante, siendo estos valores muy semejantes en cada prueba realizada. En la Figura 51 en el literal A se observa la carga suministrada en la batería totalmente desgastada, en el literal B se muestra la carga completa, cabe recalcar que las baterías solo se deben cargar las dos celdas según el indicador de carga, ya que se utilizó baterías de 7.4 V.

**Figura. 51**

*Carga de baterías*



**Fuente.** Autor

## **b) Comunicación y distancia**

Durante las pruebas se observó que la comunicación mediante Shield LoRa es fluida y constante, cabe recalcar que para utilizar el prototipo primero se debe encender el vehículo y luego el guante háptico con el fin de establecer la conexión entre ambos dispositivos, se realizó pruebas de distancia verificando que no existió interferencias o pérdidas de comunicaciones. A continuación, se detalla en la Tabla 15 las pruebas de distancias de comunicación realizadas, no se hizo uso extendido en las distancias ya que, el principal objetivo es la retroalimentación de fuerza del guante háptico.

**Tabla 15**

*Pruebas de distancia y comunicación*

| Distancia  | Parámetros   |                                  |                          |                             |
|------------|--------------|----------------------------------|--------------------------|-----------------------------|
|            | Comunicación | Función de todos los componentes | Perdidas de comunicación | Retrasos en la comunicación |
| 10 metros  | Correcta     | SI                               | NO                       | NO                          |
| 30 metros  | Correcta     | SI                               | NO                       | NO                          |
| 50 metros  | Correcta     | SI                               | NO                       | NO                          |
| 100 metros | Correcta     | SI                               | NO                       | NO                          |

Fuente. Autor

**c) Pruebas a uso continuo sin obstáculos existentes**

En esta sección se controló el vehículo de manera continua hasta agotar su batería por el lapso de 1 hora y 23 minutos alrededor de la cancha de pruebas sin obstáculos existentes, durante la ejecución de la prueba no existió inconvenientes en el desempeño de los dispositivos, se tomó como parámetro principal de prueba la dirección, manejo y actuación de sensores de retroalimentación del guante. En la Figura 52 se muestra la prueba de dirección y manejo en movimiento continuo.

**Figura. 52**

*Prueba de uso continuo sin obstáculos*



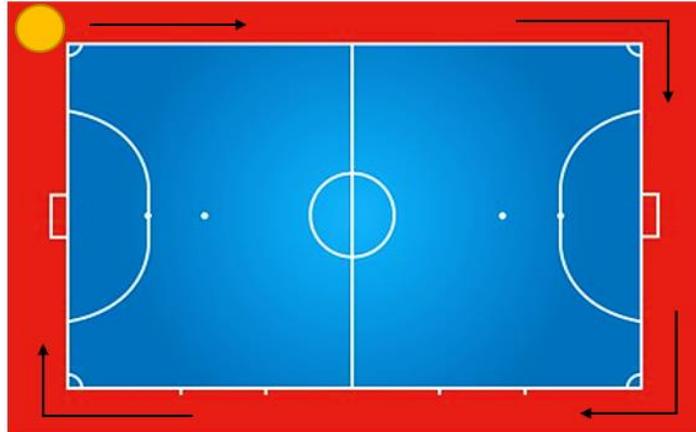
Fuente. Autor

## Ruta establecida para la prueba

En la Figura 53 indica la ruta que fue establecida para el desempeño del vehículo y el guante háptico con retroalimentación de fuerza en la prueba.

**Figura. 53**

*Ruta prueba uso continuo sin obstáculos*



**Fuente.** Autor

### d) Pruebas de detección de obstáculos con retroalimentación de fuerza y vibración

En esta sección se volvió a cargar la batería y se colocó varios obstáculos para que el vehículo detecte y envíe la información al guante háptico, como se detalla en la Figura 54.

**Figura. 54**

*Obstáculos existentes en la cancha de pruebas*



**Fuente.** Autor

La prueba realizada con el fin de comprobar la fuerza del exoesqueleto evitando que cerremos la mano y también activando el motor vibrador asignado en la programación de cada uno de los dedos, cada sensor Sharp activa o desactiva un módulo motor vibrador que se encuentra ubicado en cada dedo del exoesqueleto siendo 0 apagado y 1 encendido con respecto al sensor que se active, avisando al usuario que existe un obstáculo y debemos de cambiar de dirección el vehículo.

Por otro lado, el dedo pulgar está programado con una respuesta PWM vibratoria de 0 a 255 que depende de la velocidad del vehículo, a mayor velocidad mayor vibración del módulo motor vibrador del dedo pulgar. A continuación, en la Figura 55 se observa el vehículo evadiendo varios obstáculos en la cancha de pruebas, de acuerdo a la retroalimentación de fuerza y vibratoria recibida del vehículo al guante.

### **Figura. 55**

*Prueba de detección de obstáculos y retroalimentación de fuerza*



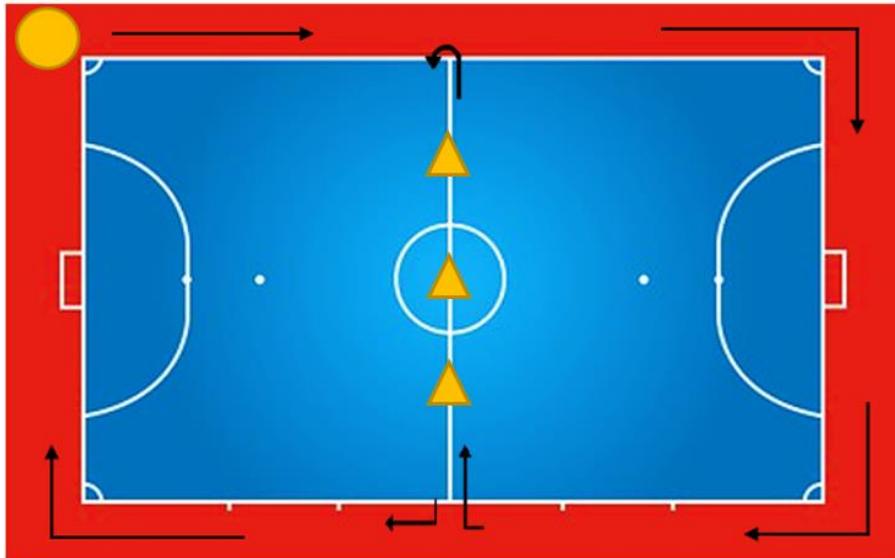
**Fuente.** Autor

### **Ruta establecida para la prueba**

En la presente ruta, a diferencia de la anterior prueba, se colocó 3 obstáculos en la mitad de la cancha y se realizó varios recorridos como se muestra en la Figura 56.

**Figura. 56**

*Ruta prueba con obstáculos existentes.*



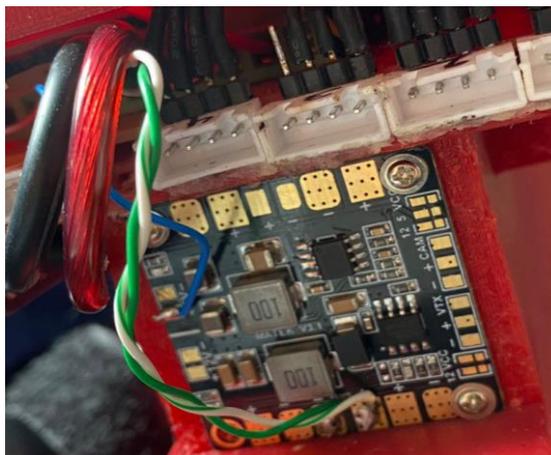
**Fuente.** Autor

**e) Prueba de actuadores**

Los actuadores del guante háptico funcionaron correctamente, pero en un inicio los servomotores tuvieron una pequeña complicación debido a que los servos soportan hasta 7.2 voltios y la batería ofrece al 100% 8.4 voltios, por lo que se optó por añadir una placa distribuidora de voltaje para evitar que los servos sufran daños, ya que la placa entrega 5 voltios, lo cual es necesario para que los servomotores funcionen correctamente como se observa en la siguiente Figura 57.

**Figura. 57**

*Montaje de placa distribuidora de voltaje*



**Fuente.** Autor

Con respecto al vehículo los actuadores funcionaron correctamente, se realizó otra prueba completamente distinta que consistió en ubicar el vehículo en una superficie semi inclinada para verificar la fuerza de los motores de movimiento y se corroboró que funciona correctamente, ya que los motorreductores Pololu tienen más torque que velocidad como se observa en la Figura 58.

**Figura. 58**

*Prueba de superficie semi inclinada*



**Fuente.** Autor

**f) Prueba de sensores Flex y giroscopio**

Para esta prueba se verificó que los sensores Flex funcionen correctamente al momento de doblar los dedos de la mano conjuntamente con el exoesqueleto, enviando una señal hacia el vehículo y controlando la velocidad del mismo. El giroscopio ubicado en el antebrazo nos ayudó a la dirección del vehículo, dando así la disposición de avanzar, retroceder, girar a la izquierda o derecha, junto a esto el encendido de las luces led respecto a cada dirección comandada desde el usuario como se observa en la Figura 59.

### **Figura. 59**

*Prueba de velocidad y direcciones del vehículo*



**Fuente.** Autor

## **3.7 Resultados**

En base a las pruebas realizadas a continuación se determina la funcionalidad que obtuvo cada componente en la ejecución de las pruebas comandadas, esto permitió verificar como fue el desempeño del guante háptico y del vehículo.

### **3.7.1 Desempeño del controlador**

En un principio para el vehículo se empezó a utilizar el Arduino Uno para el desarrollo del prototipo, pero se tuvo un problema con este controlador, ya que por el uso de la Shield LoRa algunos pines quedaban inhabilitados, por lo que se optó por reemplazar el Arduino Uno por un Arduino Mega en el cual no se presentaba este inconveniente.

En lo que respecta al guante háptico se utilizó un Arduino Mega desde el inicio por la gran cantidad de sensores y motores a controlar, por lo que se obtuvo buenos resultados al momento de realizar la ejecución de los cálculos tanto de los sensores Sharp, sensores Flex y giroscopio. Los parámetros que se consideraron para las pruebas del guante háptico y el vehículo se detallan en la Tabla 16.

**Tabla 16***Resultados de la ejecución del controlador*

| <b>Parámetros</b>                             | <b>Resultados</b> |
|---|-------------------|
| <b>Conexiones</b>                             | Correctas         |
| <b>Envío de datos</b>                         | Correctas         |
| <b>Recepción de datos</b>                     | Correctas         |
| <b>Velocidad de respuesta a instrucciones</b> | Adecuada          |

**Fuente.** Autor**3.7.2 Desempeño de los actuadores**

En el guante háptico se tiene 2 tipos de actuadores los cuales son los servomotores y los módulos motor vibrador para ello en el transcurso de las pruebas se observó que los módulos motor vibrador y servomotores funcionaban correctamente sin ningún inconveniente, se verificó que a la hora de la ejecución de los parámetros programados lo realizaba de una manera rápida y sin problemas, también se observó que en ocasiones existía sobrecalentamiento en las partes, debido a que el funcionamiento era continuo, sí presentaba algunas perturbaciones o vibraciones entre otras para lo cual se realizó una descripción en la Tabla 17.

**Tabla 17***Resultados de la ejecución de los actuadores*

| <b>Parámetros</b>              | <b>Resultados</b>                          |  |                                      |                                  |
|--------------------------------|--|--|--------------------------------------|----------------------------------|
|                                | <b>Motor Vibrador<br/>(Guante Háptico)</b> | <b>Servomotor<br/>(Guante Háptico)</b> | <b>Motores Pololu<br/>(Vehículo)</b> | <b>Servomotor<br/>(Vehículo)</b> |
| <b>Conexiones</b>              | Correctas                                  | Correctas                              | Correctas                            | Correctas                        |
| <b>Voltaje de operación</b>    | Correcto                                   | Correcto                               | Correcto                             | Correcto                         |
| <b>Ensamblaje</b>              | Correcto                                   | Correcto                               | Correcto                             | Correcto                         |
| <b>Funcionamiento continuo</b> | Si   | Si                                     | Si                                   | Si                               |
| <b>Sobrecalentamientos</b>     | Bajo                                       | Bajo                                   | Bajo                                 | No                               |
| <b>Vibraciones</b>             | No   | Bajo                                   | No                                   | No                               |

**Fuente.** Autor

En la Tabla 17 se describen los resultados que se valoró para verificar el rendimiento de los motores Pololu y el servomotor del vehículo, también los servomotores y motor vibradores del guante háptico, concluyendo que los servomotores del guante háptico tienen

una leve vibración debido a la comunicación de datos, también que el prototipo cumple con los requerimientos planteados a pesar de tener un ligero calentamiento en algunos actuadores, ya que la reacción de respuesta es eficiente a la hora de su funcionamiento. En la Figura 60 se observa la actuación del vehículo con respecto a la respuesta del guante Háptico.

**Figura. 60**

*Actuación de los actuadores*



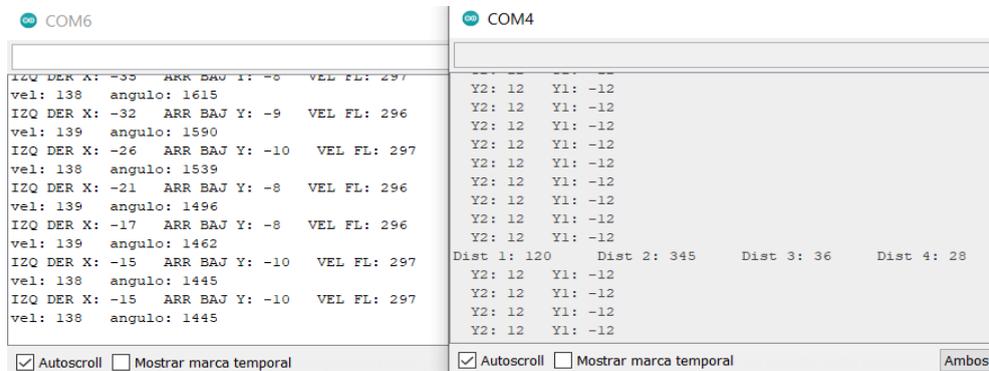
**Fuente.** Autor

### 3.7.3 Desempeño de la comunicación

Para la comunicación de los Shield LoRa se tomó el tiempo que tarda en establecer conexión entre los dos dispositivos, este tiempo variaba en cada prueba realizada siendo de 1 a 3 segundos. Una vez enlazado, la comunicación fue fluida, a distancias cercanas no hubo retrasos ni desconexiones. Se realizó distintas pruebas para verificar la distancia de comunicación y se alejó ambos dispositivos hasta 100 metros de distancia como se muestra anteriormente en la Tabla 15, la comunicación de los módulos LoRa no presento problema alguno, se tuvo énfasis en la transmisión y recepción de datos de forma bidireccional como se observa en la Figura 61.

**Figura. 61**

*Comunicación bidireccional Shield LoRa*



**Fuente. Autor**

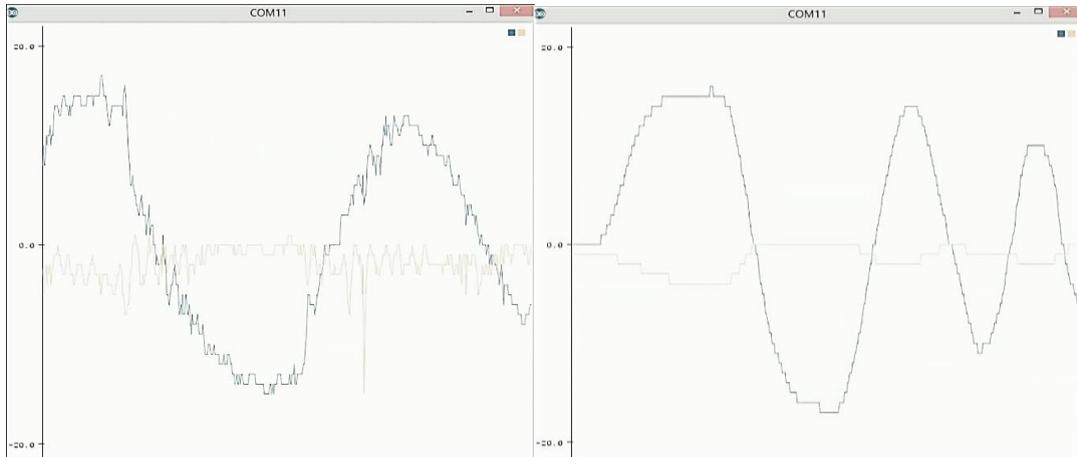
En la figura 61 se observa que existe una comunicación de datos mediante la interfaz de Arduino, donde la pantalla COM6 nos muestra los datos que recibe el vehículo y la pantalla COM4 los datos que obtiene el guante háptico. La pantalla COM6 nos muestra la dirección de los ejes del giroscopio conjuntamente con los ángulos de inclinación, la pantalla COM4 nos muestra la distancia de los sensores Sharp. Mediante esta Figura podemos observar que la comunicación bidireccional es efectiva y cumple con los parámetros establecidos.

### 3.7.4 Desempeño del giroscopio

El giroscopio en un inicio trabajó con los datos obtenidos directamente, pero presentaba muchos picos y distorsiones, para lo que se integró en la programación un filtro de datos para que esta señal mejore, eliminando los picos en las lecturas y reduciendo el ruido, mejorando la señal y los datos obtenidos como se observa en la Figura 62.

## Figura. 62

### Mediciones del giroscopio sin filtros y con filtros



Fuente. Autor

Como se observa en la Figura 62, la imagen de la izquierda indica las mediciones del giroscopio sin filtros en la que se identifica muchas perturbaciones por las vibraciones con movimientos que se generan en el giroscopio, mientras que en la imagen de la derecha se observa una gráfica sin perturbaciones y esto permite una estabilidad para el guante háptico.

### 3.7.5 Desempeño de los sensores

Para el funcionamiento de los sensores de distancia marca Sharp del vehículo se realizó en la programación un promedio de 20 muestras o mediciones, para reducir datos erróneos y no exista desviaciones que interfieran con el recorrido del vehículo. En la Figura 63 se muestra la prueba realizada para verificar el desempeño de los sensores de distancia con obstáculos y en distintos terrenos de manejo.

**Figura. 63**

*Manejo del vehículo en distintos terrenos*



**Fuente.** Autor

Para el funcionamiento de los sensores Flex al momento de flexionar los dedos de la mano estos varían sus valores y por medio de un mapeo se realiza el control de la velocidad del vehículo donde a mayor dobles de los dedos más velocidad y viceversa como se observa en la Figura 63.

### **3.7.6 Duración de la batería**

Mediante el cálculo de la selección de las baterías se había establecido una batería de 1.1 amperios hora y otra batería de 2.2 amperios hora para un tiempo de 1 hora, las baterías seleccionadas en el transcurso de las pruebas no presentaron deficiencia alguna.

La batería del vehículo que es de 1100 mA/h sobrepasó el tiempo de uso sin presentar problemas al tiempo establecido (1 hora) al igual que la batería del guante háptico de 2.2 A/h, esto se debe a que se sobrepasó la corriente que se necesitaba acorde a los cálculos realizados, ya que fueron esos valores de corriente que se encontró comercialmente. Los parámetros a considerar fueron los que se muestran en la Tabla 18. Cabe recalcar que las baterías cargadas al 100% nos da un valor máximo de 8.4 V.

**Tabla 18***Resultados de las baterías*

| <b>Parámetros</b>                                       | <b>Batería 1<br/>(Vehículo)</b>   | <b>Batería 2<br/>(Guante)</b>     |
|---|-----------------------------------|-----------------------------------|
| <b>Carga Total</b>                                      | 8.2 V                             | 8.4 V                             |
| <b>Tiempo de uso</b>                                    | (1 hora 23 minutos)<br>83 minutos | (1 hora 23 minutos)<br>83 minutos |
| <b>Tensión de las celdas al<br/>finalizar la prueba</b> | 4.2 V                             | 4.3 V                             |

**Fuente.** Autor

## CONCLUSIONES

- La descripción detallada de las tecnologías relacionadas con esta implementación proporciona una visión general de los adelantos tecnológicos de este campo y destaca la importancia de la retroalimentación háptica en la interacción humano-máquina. A medida que la tecnología avanza, se espera que la implementación de un guante háptico con retroalimentación de fuerza siga siendo una herramienta valiosa para la sociedad
- El diseño del guante háptico con retroalimentación de fuerza es un proceso que involucra la integración de diversas tecnologías y la consideración de múltiples factores, como la ergonomía, la eficiencia energética y la capacidad de respuesta háptica. A través de la investigación y el desarrollo de este proyecto, se pudo demostrar que el diseño de un guante háptico con retroalimentación de fuerza es factible y puede mejorar significativamente la experiencia del usuario en diversas aplicaciones, como la simulación de entornos virtuales y la teleoperación de robots.
- Durante las pruebas y resultados de la implementación del guante háptico con retroalimentación de fuerza se concluyó que la retroalimentación de fuerza es exitosa conjuntamente con la activación y desactivación de los distintos sensores con respecto a los obstáculos que presenta el vehículo, dando una respuesta vibratoria y el control total al usuario.
- El guante háptico con retroalimentación de fuerza se tuvo que dividir en 3 secciones para un mejor acople y maniobrabilidad para el usuario, siendo estas; el guante elástico, el exoesqueleto y la zona de control montada en el antebrazo.
- A mayores elementos electrónicos a comandar el guante háptico se hace más complejo y necesita de un controlador que brinde mayores recursos, pero con respecto a la transmisión de datos y ejecución de acciones es completamente eficaz en su totalidad.

## RECOMENDACIONES

- Explorar las nuevas técnicas de retroalimentación háptica que puedan mejorar más la experiencia del usuario. Además, se podría considerar el diseño de guantes hápticos con retroalimentación de fuerza más ergonómicos, eficientes y personalizados para acoplarse a las necesidades específicas de los usuarios finales.
- El diseño mecánico y electrónico del guante háptico puede ser reducido e impreso mediante una escala menor en los archivos de impresión 3D para el uso de un niño o una mano más pequeña, también se recomienda hacer uso del modo espejo en las piezas para poder integrar un guante funcional acorde a la mano izquierda del usuario, teniendo en cuenta que los dispositivos que integran el mismo también pueden ser sustituidos por unos más pequeños para que el prototipo no sea muy voluminoso y pesado, esto con el fin de reducir costos, tamaño, peso, mejorando la calidad de los elementos pertenecientes.
- Se recomienda buscar tecnologías con otro tipo de retroalimentación para obtener una mayor interacción entre hombre máquina, también puede ser mejorado con elementos tecnológicos como: sensores fotoeléctricos, sensores y cámaras de movimiento o dispositivos de realidad virtual.
- Al momento de diseñar en 3D se recomienda tener en cuenta cada detalle de distancia y movilidad puesto que esto permitirá que el diseño sea lo más semejante a la realidad a la hora de imprimirlo y permitiendo la comodidad del usuario.

## Bibliografía

- Abdi, E. K. (2020). Haptics in Teleoperated Medical Interventions: Force Measurement, Haptic Interfaces and Their Influence on User's Performance. . *IEEE Transactions on Biomedical Engineering.*, 3438-3451.
- Acosta, N. (2018, Diciembre 14). *CUIDA TU DINERO*. Retrieved from CUIDA TU DINERO: <https://www.cuidatudinero.com/13176145/tipos-de-maquinas-cnc>
- Altamirano-Droguett, J. E. (2019). La simulación clínica . *Revista Electrónica Educare*, 5(2), 167-187.
- Amalialia. (2020, Enero). *WordPress.com*. Retrieved from <https://amalialia2.wordpress.com/2020/01/30/tipos-de-arduino/>
- Andres, A. (2017). mplementación de una herramienta virtual mediante un dispositivo háptico para la rehabilitación de pacientes con daño cerebral adquirido. (Tesis de Grado). Universidad de las fuerzas armadas ESPE , Latacunga, Cotopaxi, Ecuador.
- Arduino . (2022). *Arduino Due*. Retrieved from Arduino Store: <https://store-usa.arduino.cc/products/arduino-due?selectedStore=us>
- Arduino . (2022). *Arduino Ethernet*. Retrieved from Arduino Store: <https://store-usa.arduino.cc/products/arduino-ethernet-shield-2?selectedStore=us>
- Arduino. (2022). *Arduino*. Retrieved Abril 10, 2022, from <https://arduinodhtics.weebly.com/tipos-de-arduino.html>
- Arri, F. (2019). Leer con las manos. Una revisión sobre los estudios hápticos. *InMediaciones de la Comunicación*,, 7(2), 151-160.
- AV Electronics*. (2019). Retrieved from <https://avelectronics.cc/producto/flex-sensor-2-2/>
- Ballesteros, S., & Martínez Niño, C. E. (2021). Experiencias de la realidad virtual háptica En la simulación de procedimientos de cateterismo cardiológico.
- Berezowsky, C. A. (2018, Octubre). Anatomical Study of the Superficial Palmar Arch, its Relation to the Kaplan Cardinal Line and Literature Review. *Revista Iberoamericana de Cirugía de la Mano*.
- Borrego González, A. (2020, Marzo). RELACIÓN DE LA PRESENCIA Y LA CORPORIZACIÓN EN REALIDAD VIRTUAL EN INDIVIDUOS CON ICTUS.
- Cableworld. (2019, noviembre 13). *Cableworld*. Retrieved from <https://www.cableworld.es/cableworld/tecnologia-haptica-que-es-para-que-sirve/>
- Carrod. (2022). *Carrod Electronica*. Retrieved Abril 1, 2022, from <https://www.carrod.mx/products/motor-reductor-6-v-200-rpm-3-v-150-rpm>
- Castro, A. (2019, Septiembre 9). *dron profesional*. Retrieved Octubre 25, 2022, from <https://dronprofesional.com/blog/tutorial-teorico-practico-con-servos-y-arduino/>

- Copadata. (2021, Junio). *Copadata*. Retrieved from <https://www.copadata.com/es/productos/zenon-software-platform/visualizacion-control/que-significa-hmi-interfaz-humano-maquina-copa-data/>
- Cruz, D. (2022, Abril 8). *Tecnosinergia*. Retrieved Abril 10, 2022, from <https://tecnosinergia.zendesk.com/hc/es/articles/360001380312--Cómo-funcionan-los-sensores-fotoeléctricos-fotoceldas-fotobeams->
- Culque Toapanta, W. V. (2022). Simulador electrónico con feedback háptico para entrenamiento pedagógico.
- Curtis, A. (2021, Diciembre 20). *Youtube*. Retrieved Mayo 12, 2022, from [https://www.youtube.com/watch?v=s1meUk\\_pSKE](https://www.youtube.com/watch?v=s1meUk_pSKE)
- Dominguez, J. (2017, Septiembre). *Areatecnologia*.
- du Plessis, T. D. (2021). *A review of active hand exoskeletons for rehabilitation and assistance*. *Robotics* (Vol. 20(2)).
- Electronica. (2022). *GY-521*. Retrieved Mayo 12, 2022, from <https://laelectronica.com.gt/giroscopio-y-acelerometro-mpu-6050-gy-521>
- Electronics AV*. (2019). Retrieved from <https://avelectronics.cc/producto/flex-sensor-2-2/>
- ElectroStore, T. o. (2022, Noviembre 20). *Grupo Electrostore*. Retrieved from <https://grupoelctrostore.com/shop/sensores/infrarrojos/sensor-sharp-gp2y0a02yk0f-20-150cm-analogico-de-distancia/>
- Enriquez, M. (2018). *Sparkfun*. Retrieved from <https://www.sparkfun.com/products/14451>
- Escobar, J., & Quiros, R. (2017). Analisis Biomecanico de las fuerzas de agarre de la mano. Bucaramanga.
- Espressif. (2022). *ESP32*. Retrieved from [Espressif: https://www.espressif.com/en/products/socs/esp32](https://www.espressif.com/en/products/socs/esp32)
- Fernandez, Y. (2020, Agosto 3). *Xataka*. Retrieved from <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- Filament2Print. (2022). *Filament2Print*. Retrieved from [https://filament2print.com/es/blog/71\\_importancia-relleno-impresion-3d.html](https://filament2print.com/es/blog/71_importancia-relleno-impresion-3d.html)
- FreeShopping. (2019, Mayo 24). *Associated CR12 Ford F-150*. Retrieved from <https://www.rcplanet.com/associated-cr12-ford-f-150-pick-up-rtr-black-asc40001/>
- Frutos, A. L. (2019). Educacion Inclusiva: Discapacidad visual. *Percepción cinestésica y tactil*. ONCE.
- G, A. (2021, Mayo 20). *Cómo utilizar un puente H con Arduino*. Retrieved from [Automatización para todos: https://www.automatizacionparatodos.com/puente-h-arduino/](https://www.automatizacionparatodos.com/puente-h-arduino/)
- Hoperf. (2022). *Microelectronics*. Retrieved from [https://www.tme.eu/en/katalog/rf-modules\\_113194/?art=RFM43-433-S1&page=11](https://www.tme.eu/en/katalog/rf-modules_113194/?art=RFM43-433-S1&page=11)

- Imrelevsa. (2022). Retrieved Mayo 12, 2022, from <http://www.velasco.com.ec/velasco/producto.php?id=4658>
- Islas Silvas, O. A. (2019). *Algoritmo inteligente para la optimización del proceso de carga en una batería LiPo*.
- Jorge López, A. (2019). Desarrollo de un guante háptico para entornos virtuales. *Repositorio Digital*.
- Kadhun, A. A. (2021, Diciembre). Implementation dc motor as servomotor by using arduino and optical rotary encoder. *Materials Today: Proceedings*.
- Keyence. (2022). *Características y ventajas de los sensores fotoeléctricos*. Retrieved from Keyence:  
<https://www.keyence.com.mx/ss/products/sensor/sensorbasics/photoelectric/feature/>
- Leibson, S. (2018, Noviembre 28). *Digi-Key*. Retrieved Mayo 3, 2020, from <https://www.digkey.com/es/articles/fundamentals-distance-measurement-gesture-recognition-tof-sensors>
- León, M. M.-F.-H. (2019). *Impresión 3D con materiales elásticos*.
- Llamas, L. (2016, Diciembre 6). *Ingeniería, informática y diseño*. Retrieved Enero 23, 2020, from Comunicación inalámbrica en arduino con módulos RF:  
<https://www.luisllamas.es/comunicacion-inalambrica-en-arduino-con-modulos-rf-433mhz/>
- Llamas, L. (2017, Noviembre 17). *Ingeniería, informática y diseño*. Retrieved Mayo 18, 2020, from <https://www.luisllamas.es/modelos-de-raspberry-pi/>
- Martinez, V. (2014, Septiembre). Introducción a la plataforma Arduino y el sensor Ultrasonico HC-SR04.
- Medellín, H. I. (2011). Desarrollo de un sistema de proyección de películas virtuales para personas invidentes. *Impulso*.
- Meier, L. (2020, Marzo 3). *Pixhawk*. Retrieved Mayo 18, 2020, from <https://pixhawk.org/products/>
- Mena Palomeque, J. &. (2017). *Diseño, implementación y control de un péndulo invertido como plataforma de ensayo para diferentes estrategias de control*.
- Mendez, M. (2018). *desarrollo de un guante háptico con retroalimentación táctil*. Tesis, Madrid. Retrieved Mayo 12, 2022
- Mera, A., & Ruano, D. (2019). *Implementación de un robot de navegación autónoma terrestre para evasión de obstáculos usando el dispositivo PIXHAWK*. Tesis, Universidad de las Américas, Quito. Retrieved Diciembre 1, 2019
- Narvaez, K., & Contreras, V. (2020). *Diseño y desarrollo de un prototipo de red de sensores IOT*. Tesis, Universidad Politécnica Salesiana, Guayaquil. Retrieved Abril 10, 2022, from <https://dspace.ups.edu.ec/bitstream/123456789/19439/1/UPS-GT003019.pdf>

- Navarro, D., Rios, L., & Parra, H. (2004). *SENSORES DE ULTRASONIDO USADOS EN ROBÓTICA MÓVIL PARA LA MEDICIÓN*. *Scientia Et Technica*, 35-40.
- Naylamp. (2022). *Mechatronics*. Retrieved Abril 1, 2022, from <https://naylampmechatronics.com/sensores-posicion-inerciales-gps/33-modulo-mpu6050-acelerometro-giroscopio-i2c.html>
- Osaka Electronics. (2022). Retrieved Mayo 12, 2022, from <https://osakaelectronicsltda.com/modulos-robotica/modulos-comunicacion/modulo-bluetooth-hc-05.html>
- Peña Cortés, C. A.-G.-G. (2022). Análisis del nivel de compromiso en la programación de dispositivos hápticos por medio de una interfaz cerebro computador. *Revista de Investigación, Desarrollo e Innovación*, 565.
- Peña-Cortés, C. A.-G.-G. (2022). Análisis del nivel de compromiso en la programación de dispositivos hápticos por medio de una interfaz cerebro computador. . *Revista de Investigación, Desarrollo e Innovación*, 565.
- Perez, A. (2020). *Robotica* . Retrieved from <http://electr3s.com/inicio/222-pwm-vibrador-arduino.html>
- Priego, L. (2022, Junio 1). ¿Por qué hay tan pocas personas zurdas en el mundo? *¿Por qué hay tan pocas personas zurdas en el mundo?*
- Prometec. (2022). *Wemos* . Retrieved from Prometec: <https://www.prometec.net/wemos-d1-esp8266-wifi/>
- Quito Bacuilima, F. V. (2022). Desarrollo de una interfaz háptica como soporte a la neurorrehabilitación de la mano derecha. *Bachelor's thesis*. Universidad del Azuay. Retrieved from redalyc.
- Ramar, J. (2014, Septiembre 26). *Arduino*. Retrieved from TecnoInfE: <https://tecnoinfo.wordpress.com/2014/09/26/arduino/>
- Rambal. (2022). *Automotización y Robótica*. (D. Sensor Flex, Producer) Retrieved Abril 10, 2022, from <https://rambal.com/presion-peso-nivel-flex/250-sensor-flex.html>
- Repullo Menéndez, M. (2021, Noviembre). Desarrollo de un guante háptico con realimentación táctil.
- Robotics Sando. (2022). *Sando Robotics*. Retrieved Mayo 12, 2022, from Sando Robotics: <https://sandorobotics.com/producto/999/>
- Robots. (2022). *Robots didacticos*. Retrieved Abril 1, 2022, from <https://robots-argentina.com.ar/didactica/servos-caracteristicas-basicas/>
- Roda, G. (2020, Mayo 18). *Arduino, Genuino, Raspberry Pi. Noticias y proyectos*. Retrieved Mayo 19, 2020, from <https://descubrearduino.com/comparamos-varias-de-las-arduino-mas-utilizadas/>

- Rodríguez Ramirez, D., Ruiz, C. E., & Torres, S. A. (2020). La mano. Aspectos anatómicos I. Generalidades, osteología y. *Morfología*.
- Rodriguez, D. (2018). *Dispositivos LoRa de comunicacion a largo alcance y bajo consumo energetico*. Tesis, Universidad Politecnica de Madrid, Madrid España. Retrieved Abril 12, 2022
- Rodriguez, L. (2022). Retrieved Myo 12, 2022, from <https://como-funciona.co/un-giroscopio/>
- S.C. Enriquez, Y. N. (2014, Septiembre). *Researchgate*. Retrieved from [http://www.ja2014.upv.es/wp-content/uploads/papers/paper\\_44.pdf](http://www.ja2014.upv.es/wp-content/uploads/papers/paper_44.pdf)
- Salazar, C. (2020, Noviembre). *GSL Industrias*.
- Sánchez, J. (2017). Incremento de inmersividad en usuarios de entornos de realidad virtual utilizando una interfaz háptica para limitación de movimiento en dedos.
- Sánchez, J. P. (2019). *Incremento de inmersividad en usuarios de entornos de realidad virtual utilizando una interfaz háptica para limitación de movimiento en dedos*.
- Sando. (2022). *Módulo Mini Vibrador PWM*. (S. Robotic, Producer) Retrieved Abril 1, 2022, from <https://sandorobotics.com/producto/sd-0041/>
- Sando Robotics. (2020). Retrieved from <https://sandorobotics.com/producto/bms-3s-10a/>
- SICNOVA. (2022). *SICNOVA*. Retrieved from [https://sicnova3d.com/blog/experiencias-3d/que-es-el-pla-en-impresion-3d-y-para-que-se-utiliza/#:~:text=El%20PLA%20\(%C3%A1cido%20polil%C3%A1ctico\)%20ofrece,tiene%20una%20deformabilidad%20muy%20baja](https://sicnova3d.com/blog/experiencias-3d/que-es-el-pla-en-impresion-3d-y-para-que-se-utiliza/#:~:text=El%20PLA%20(%C3%A1cido%20polil%C3%A1ctico)%20ofrece,tiene%20una%20deformabilidad%20muy%20baja).
- SUNLU. (2022). Retrieved from [https://www.sunlu.com/?sca\\_ref=128229.Ttc0dFOifj](https://www.sunlu.com/?sca_ref=128229.Ttc0dFOifj)
- Techmake. (2018). Retrieved from <https://techmake.com/products/enebat00568>
- Teroson, L. (2018). *Blog profesionales del taller*. Retrieved from <https://blog.reparacion-vehiculos.es/coches-impresoras-3d>
- TRACTUS3D. (2022). *TRACTUS3D*. Retrieved from <https://tractus3d.com/es/materials/petg/#:~:text=El%20nivel%20de%20durabilidad%20del,sometidos%20a%20una%20tensi%C3%B3n%20mec%C3%A1nica>.
- UnitElectronics. (2022). *Wemos D1*. Retrieved from UnitElectronics: <https://uelectronics.com/producto/wemos-d1-wifi-esp8266-esp-12f-compatible-con-arduino/>
- Valenzuela Urrutia, D. R. (2016). Sistema de teleoperación basado en interfaz háptica.
- Vivas, A. (2018). *Interfa háptica tipo guante con retroalimentacion vibratoria*. Artículo, Colombia. Retrieved Mayo 12, 2022, from <file:///C:/Users/hp/Downloads/718-1987-1-PB.pdf>
- WAVGAT. (2022). *WAVGAT*. Retrieved from <https://www.aliexpress.com/item/4001105619912.html>

# Anexos

## Anexo A: Datos de la placa de distribución de voltaje PDB



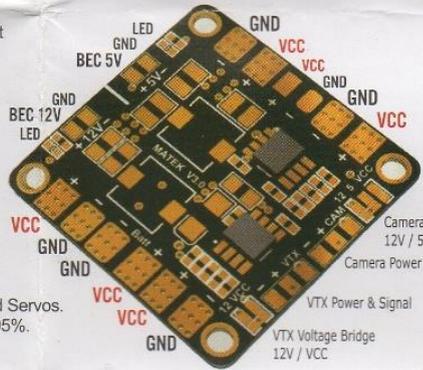
**MANUAL**

**MINI POWER HUB**  
w/ BEG 5V & 12V

SKU: HUB5V12V

The Mini POWER HUB has been engineered to provide the highest possible performance and reliability in a 36\*36mm & 4 layers PCB. The HUB distributes power from a battery pack to 4 ESCs, as well as providing synchronised & regulated DC outputs for powering cameras, servos, video transmitters, LEDs, etc. Batteries may range from 3S to 6S LiPo. The HUB also features convenient solder bridge of various voltage for camera & video TX.

**Layout**



**Features**

- 2oz copper & 4-layers PCB.
- ESC & Battery solder tabs in pairs
- Solder Bridges of various voltage for Camera & VTX.
- 5V & 12V Output LED indicators & Short circuit tolerant

**Specifications**

**General:**

- Input voltage range (3S-6S operation): 9 - 26VDC
- Regulated 5V and 12V outputs
- LED power indicators ( 5V & 12V outputs )
- 4 ESC outputs

**ESC outputs:**

- Continuous current: 20A per output
- Peak current (10 seconds/minute): 25A per output

**BEC 5V output:**

- Designed for RC Receivers, Flight controllers, OSD, and Servos.
- DC/DC synchronous buck regulator, Efficiency is up to 95%.
- Voltage: 5.0 +/- 0.1VDC
- Continuous current: 3 Amps
- Short-circuit tolerant (10 seconds/minute)
- LED(indicates voltage is in regulation)

**BEC 12V output:**

- The battery should be 4S-6S LiPo
- Standard output designed to power cameras and video transmitters, etc.
- Voltage: 12.0+/-0.2VDC
- Continuous current: 2 Amps(Max. 3A 10s/minute)
- Short-circuit tolerant(10 seconds/minute)
- LED(indicates voltage is in regulation)

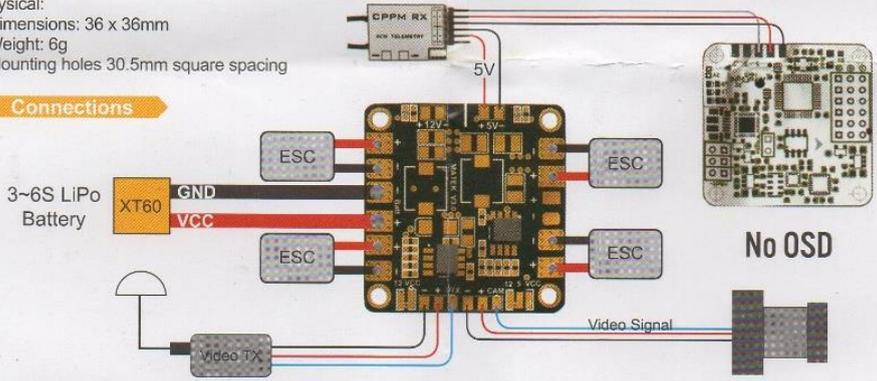
**CAM & VTX voltage bridge**

- 3S operation, choose "VCC" for 12V camera & 12V VTX. VCC=Battery Voltage.
- 4-6S operation, choose "12" for 12V camera & 12V VTX
- 3-6S operation, choose "5" for 5V camera
- Video pass through from the "CAM" pad to the "VTX" pad

**Physical:**

- Dimensions: 36 x 36mm
- Weight: 6g
- Mounting holes 30.5mm square spacing

**Connections**



**No OSD**



e.g. choose 5V for camera, 12V for VTX

## Anexo B: Datasheet MPU6050

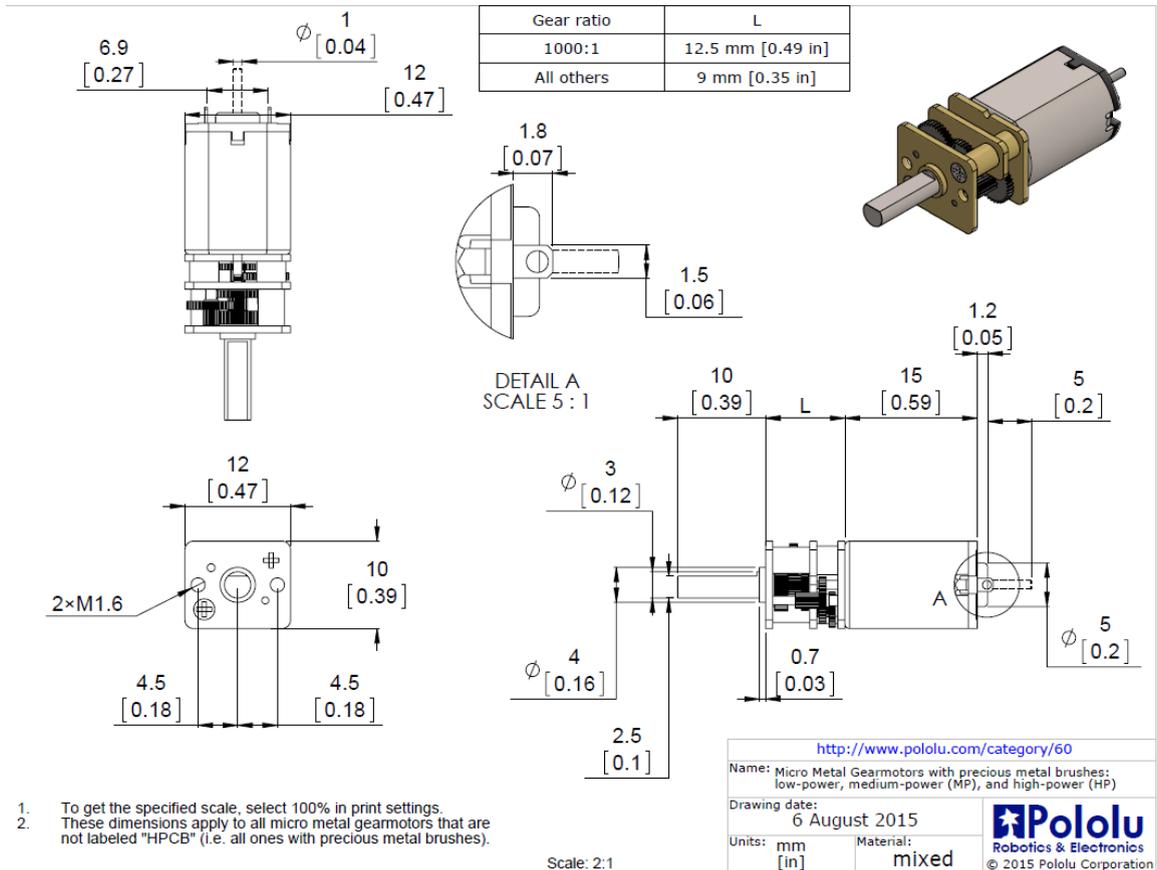
|   |  |  |
|---|--|--|
|  | <b>MPU-6000/MPU-6050 Product Specification</b> | Document Number: PS-MPU-6000A-00<br>Revision: 3.3<br>Release Date: 5/16/2012 |
|---|--|--|

### 6.9 Absolute Maximum Ratings

Stress above those listed as “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these conditions is not implied. Exposure to the absolute maximum ratings conditions for extended periods may affect device reliability.

| Parameter  | Rating                              |
|--|-------------------------------------|
| Supply Voltage, VDD  | -0.5V to +6V                        |
| VLOGIC Input Voltage Level (MPU-6050)                          | -0.5V to VDD + 0.5V                 |
| REGOUT   | -0.5V to 2V                         |
| Input Voltage Level (CLKIN, AUX_DA, AD0, FSYNC, INT, SCL, SDA) | -0.5V to VDD + 0.5V                 |
| CPOUT (2.5V ≤ VDD ≤ 3.6V )                                     | -0.5V to 30V                        |
| Acceleration (Any Axis, unpowered)                             | 10,000g for 0.2ms                   |
| Operating Temperature Range                                    | -40°C to +105°C                     |
| Storage Temperature Range                                      | -40°C to +125°C                     |
| Electrostatic Discharge (ESD) Protection                       | 2kV (HBM);<br>200V (MM)             |
| Latch-up   | JEDEC Class II (2), 125°C<br>±100mA |

## Anexo C: Datasheet Pololu



1. To get the specified scale, select 100% in print settings.
2. These dimensions apply to all micro metal gearmotors that are not labeled "HPCB" (i.e. all ones with precious metal brushes).

## Anexo D: Datasheet shield LoRa

| Symbol  | Description   | Conditions  | Min              | Typ                   | Max              | Unit                 |
|---------|---|---|------------------|-----------------------|------------------|----------------------|
| IDDSL   | Supply current in Sleep mode                            |   | -                | 0.2                   | 1                | µA                   |
| IDDIDLE | Supply current in Idle mode                             | RC oscillator enabled   | -                | 1.5                   | -                | µA                   |
| IDDST   | Supply current in Standby mode                          | Crystal oscillator enabled  | -                | 1.6                   | 1.8              | mA                   |
| IDDFS   | Supply current in Synthesizer mode                      | FSRX  | -                | 5.8                   | -                | mA                   |
| IDDR    | Supply current in Receive mode                          | <i>LnaBoost</i> Off, band 1<br><i>LnaBoost</i> On, band 1<br>Bands 2&3  | -<br>-<br>-      | 10.8<br>11.5<br>12.0  | -<br>-<br>-      | mA                   |
| IDDT    | Supply current in Transmit mode with impedance matching | RFOP = +20 dBm, on PA_BOOST<br>RFOP = +17 dBm, on PA_BOOST<br>RFOP = +13 dBm, on RFO_LF/HF pin<br>RFOP = +7 dBm, on RFO_LF/HF pin | -<br>-<br>-<br>- | 120<br>87<br>29<br>20 | -<br>-<br>-<br>- | mA<br>mA<br>mA<br>mA |

## Anexo E: Datasheet Sharp

# SHARP

GP2Y0A21YK0F

### ■ Absolute Maximum Ratings (T<sub>a</sub>=25°C, V<sub>CC</sub>=5V)

| Parameter               | Symbol           | Rating                       | Unit |
|-------------------------|------------------|------------------------------|------|
| Supply voltage          | V <sub>CC</sub>  | -0.3 to +7                   | V    |
| Output terminal voltage | V <sub>O</sub>   | -0.3 to V <sub>CC</sub> +0.3 | V    |
| Operating temperature   | T <sub>opr</sub> | -10 to +60                   | °C   |
| Storage temperature     | T <sub>stg</sub> | -40 to +70                   | °C   |

### ■ Electro-optical Characteristics (T<sub>a</sub>=25°C, V<sub>CC</sub>=5V)

| Parameter                   | Symbol          | Conditions   | MIN. | TYP. | MAX. | Unit |
|-----------------------------|-----------------|--|------|------|------|------|
| Average supply current      | I <sub>CC</sub> | L=80cm (Note 1)  | —    | 30   | 40   | mA   |
| Distance measuring          | ΔL              | (Note 1)   | 10   | —    | 80   | cm   |
| Output voltage              | V <sub>O</sub>  | L=80cm (Note 1)  | 0.25 | 0.4  | 0.55 | V    |
| Output voltage differential | ΔV <sub>O</sub> | Output voltage difference between L=10cm and L=80cm (Note 1) | 1.65 | 1.9  | 2.15 | V    |

\* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27•white face, reflectance; 90%)

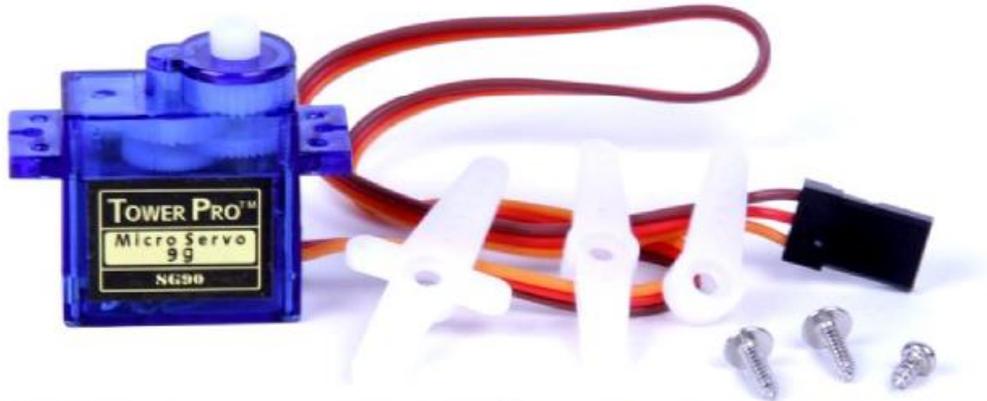
### ■ Recommended operating conditions

| Parameter      | Symbol          | Rating     | Unit |
|----------------|-----------------|------------|------|
| Supply voltage | V <sub>CC</sub> | 4.5 to 5.5 | V    |

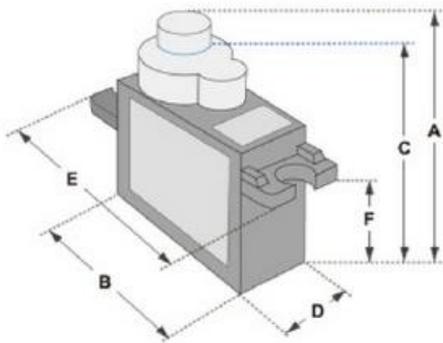
## Anexo F: Datasheet Servomotor

### SERVO MOTOR SG90

### DATA SHEET



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



| Dimensions & Specifications |         |
|-----------------------------|---------|
| A (mm) :                    | 32      |
| B (mm) :                    | 23      |
| C (mm) :                    | 28.5    |
| D (mm) :                    | 12      |
| E (mm) :                    | 32      |
| F (mm) :                    | 19.5    |
| Speed (sec) :               | 0.1     |
| Torque (kg-cm) :            | 2.5     |
| Weight (g) :                | 14.7    |
| Voltage :                   | 4.8 - 6 |

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.

PWM=Orange (⏏)  
 Vcc = Red (+)  
 Ground=Brown (-)



## Anexo G: Datasheet sensor Flex



# FLEX SENSOR FS

Special Edition Length

## Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
  - Robotics
  - Gaming (Virtual Motion)
  - Medical Devices
  - Computer Peripherals
  - Musical Instruments
  - Physical Therapy
- Simple Construction
- Low Profile

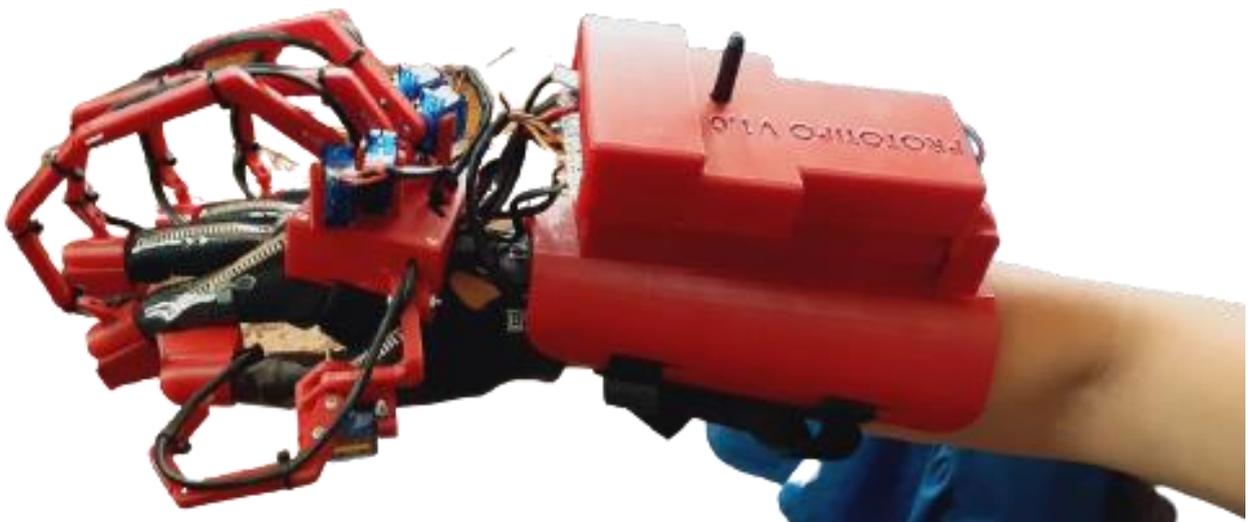
## Mechanical Specifications

- Life Cycle: >1 million
- Height:  $\leq 0.43\text{mm}$  (0.017")
- Temperature Range:  $-35^{\circ}\text{C}$  to  $+80^{\circ}\text{C}$

## Electrical Specifications

- Flat Resistance:  $25\text{K Ohms} \pm 30\%$
- Bend Resistance: minimum 2 times greater than the flat resistance at  $180^{\circ}$  pinch bend (see "How it Works" below)
- Power Rating : 0.5 Watts continuous; 1 Watt Peak

## Anexo H: Prototipo de guante háptico con retroalimentación de fuerza



## **Anexo I: Manual técnico**

### **INTRODUCCIÓN**

La implementación de cualquier tipo de prototipo conlleva la realización de un manual técnico con el fin de detallar todas las conexiones de los dispositivos utilizados para que cualquier usuario que tenga acceso al prototipo, se facilite al momento de conectar y programar. Para ello se detalla las conexiones, montaje y programación del prototipo del guante háptico con retroalimentación de fuerza.

#### **1. Conexiones de actuadores**

Para las conexiones de los actuadores se tiene una parte que va montado en la estructura del vehículo y otra en el exoesqueleto del guante háptico.

##### **a) Conexiones de los actuadores del vehículo**

Para la colocación de los componentes electrónicos del vehículo se realizó el montaje de los motores Pololu con su respectivo driver y un servomotor para la dirección de giro acorde al diagrama de la Figura 1 cabe recalcar que la batería alimenta tanto al Arduino Mega como a al servomotor y motorreductores Pololu con el fin de evitar el ruido que se genera.

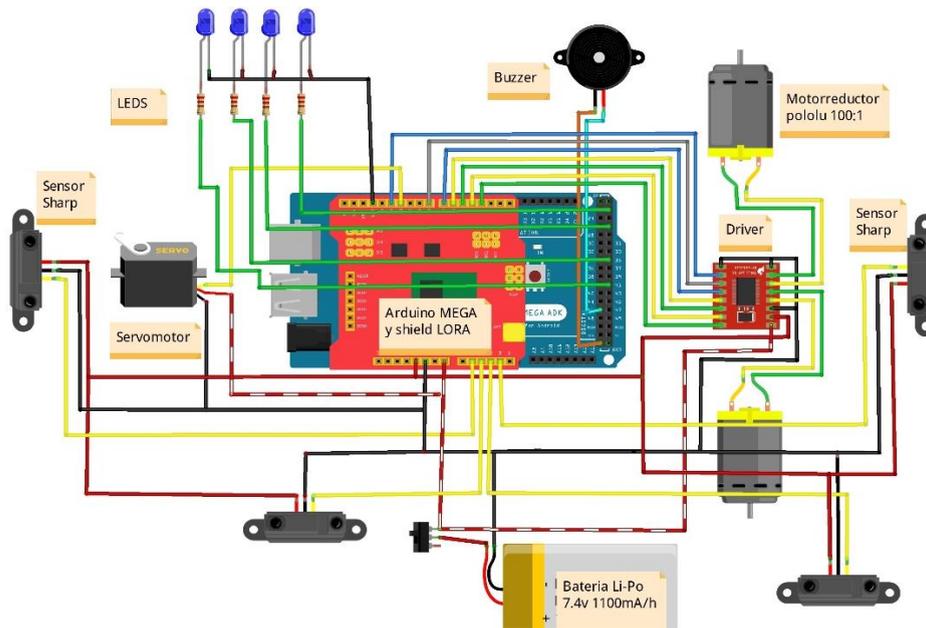


Figura. 64 Conexiones del servomotor y motorreductores Pololu

Fuente. Autor

Para las conexiones del servomotor y motores Pololu se muestra en la Tabla 1 los pines que tienen los componentes y su conexión entre ellos.

Tabla 19 Conexiones de los actuadores del vehículo

| <b>Servomotor</b>          | <b>Conexiones</b>     |
|----------------------------|-----------------------|
| VCC                        | 7v (Batería)          |
| GND                        | GND (Batería)         |
| SEÑAL                      | Pin 5 (Arduino Mega)  |
| <b>Driver motor Pololu</b> | <b>Conexiones</b>     |
| PWMA                       | Pin 3 (Arduino Mega)  |
| AIN2                       | Pin 4 (Arduino Mega)  |
| AIN1                       | Pin 5 (Arduino Mega)  |
| STBY                       | Pin 6 (Arduino Mega)  |
| BIN1                       | Pin 7 (Arduino Mega)  |
| BIN2                       | Pin 8 (Arduino Mega)  |
| PWMB                       | Pin 12 (Arduino Mega) |
| GND                        | GND (Arduino Mega)    |
| VIN                        | 7v (Batería)          |
| VCC                        | 5v (Arduino Mega)     |
| A1                         | Motor 1 primero cable |
| A2                         | Motor 1 segundo cable |

|               |                                  |
|---------------|----------------------------------|
| B2            | Motor 2 primero cable            |
| B1            | Motor 2 segundo cable            |
| <b>Buzzer</b> | <b>Conexiones</b>                |
| GND           | GND (Arduino Mega)               |
| SEÑAL         | Pin 46 (Arduino Mega)            |
| <b>Leds</b>   | <b>Conexiones</b>                |
| GND           | Gnd (Arduino Mega)               |
| SEÑAL         | Pin 23, 27,35, 41 (Arduino Mega) |

Fuente. Autor

### b) Conexión de los actuadores en el guante háptico

Para la colocación de los componentes electrónicos del guante háptico se realizó el montaje de los módulos motores vibradores y servomotores en cada dedo acorde al diagrama de la Figura 2 cabe recalcar que la batería alimenta tanto al Arduino Mega como a los servomotores y motor-vibradores con el fin de evitar el ruido que se genera.

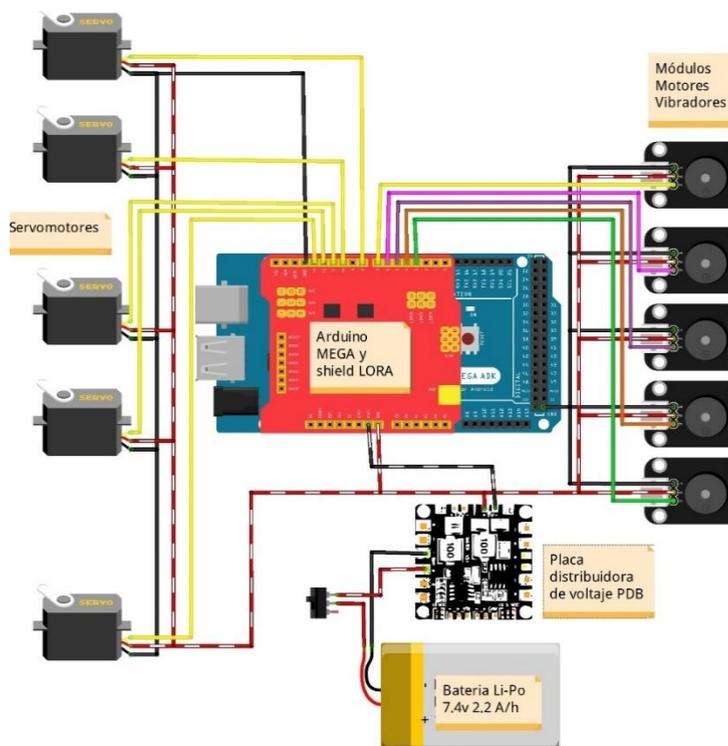


Figura. 65 Conexiones de los servomotores y motor-vibradores

Fuente. Autor

Para las conexiones de los servomotores y motor vibradores se muestra en la Tabla 2 los pines que tienen los componentes y su conexión entre ellos.

*Tabla 20 Conexiones de los actuadores del guante háptico*

| <b>Servomotor</b>     | <b>Conexiones</b>                    |
|-----------------------|--------------------------------------|
| VCC                   | 5v (PDB)                             |
| GND                   | GND (PDB)                            |
| SEÑAL                 | Pin 13, 12, 11, 10, 8 (Arduino Mega) |
| <b>Motor vibrador</b> | <b>Conexiones</b>                    |
| VCC                   | 5v (PDB)                             |
| GND                   | GND (PDB)                            |
| SEÑAL                 | Pin 7, 6, 5, 4, 3 (Arduino Mega)     |

Fuente. Autor

## 2. Conexiones de sensores

Para las conexiones de los sensores se tiene una parte que va montado en la estructura del vehículo que son los sensores Sharp de distancia y otra en el exoesqueleto del guante háptico que son los sensores Flex y el giroscopio.

### a) Conexiones de los sensores del vehículo

Para el montaje se procedió a realizar las conexiones de los cuatro sensores Sharp en puntos específicos para que el vehículo detecte los obstáculos para ello se siguió el siguiente diagrama que se muestra en la Figura 3.

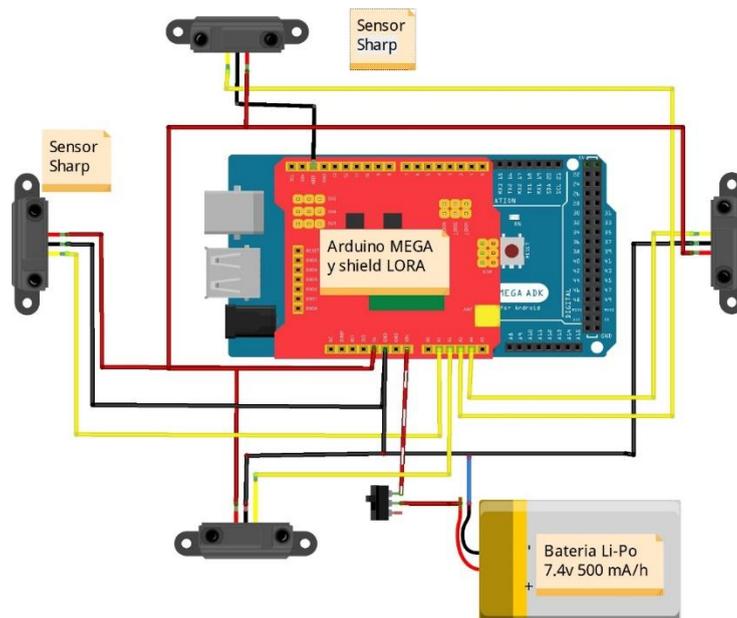


Figura. 66 Conexiones de los sensores Sharp de distancia

Fuente. Autor

Para las conexiones de los sensores Sharp se muestra en la Tabla 3 los pines que tienen los componentes y su conexión entre ellos.

Tabla 21 Conexiones de los sensores Sharp del vehículo

| Sensor Sharp | Conexiones                        |
|--------------|-----------------------------------|
| POSITIVO +   | 5v (Arduino Mega)                 |
| NEGATIVO -   | GND (Arduino Mega)                |
| SEÑAL        | Pin A1, A2, A3, A4 (Arduino Mega) |

Fuente. Autor

#### a) Conexiones de los sensores del guante háptico

Para el montaje se procedió a realizar las conexiones de los sensores Flex y el giroscopio para la colocación de los sensores Flex se realizó el montaje en un guante elástico para poder tener mejor respuesta de los sensores para ello se siguió el siguiente diagrama que se muestra en la Figura 4.

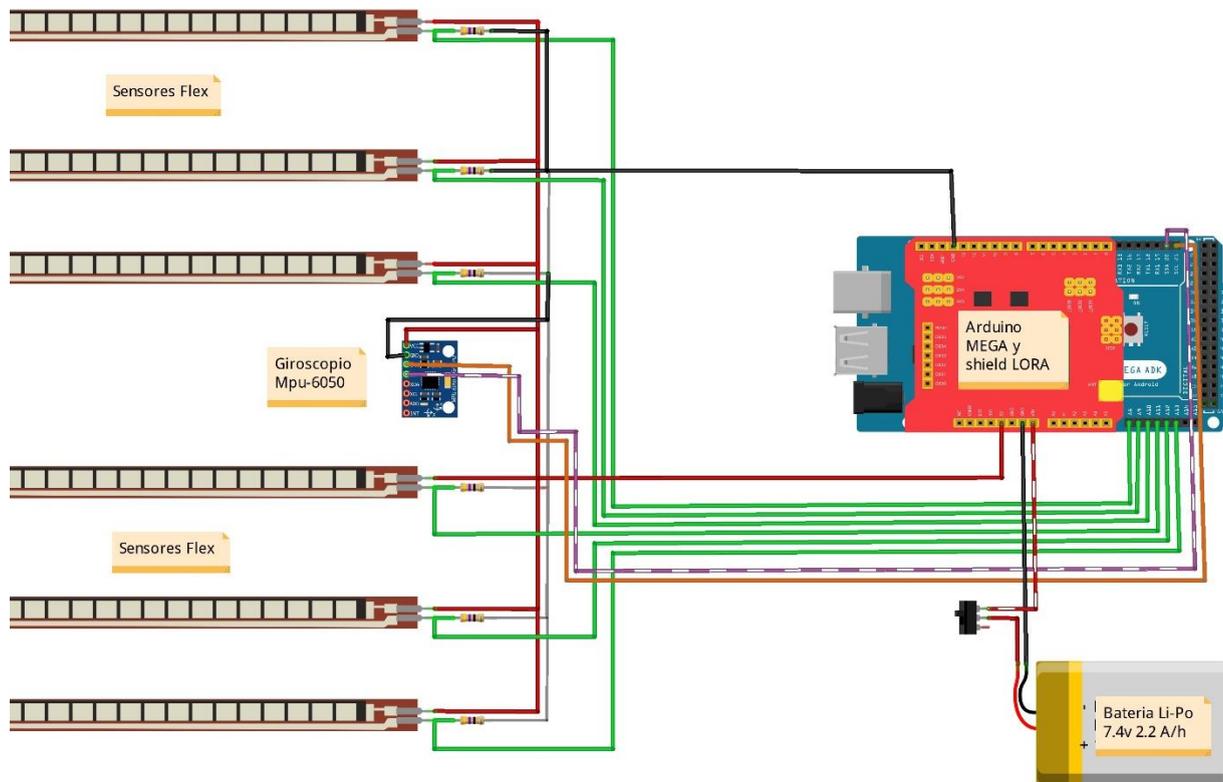


Figura. 67 Conexiones de los sensores Flex y giroscopio

Fuente. Autor

Para las conexiones de los sensores Flex y giroscopio se muestra en la Tabla 4 los pines que tienen los componentes y su conexión entre ellos.

Tabla 22 Conexiones de los sensores Sharp y giroscopio del guante háptico

| Sensor Flex | Conexiones                                    |
|-------------|---|
| POSITIVO +  | 5v (Arduino Mega)                             |
| NEGATIVO -  | GND (Arduino Mega)                            |
| SEÑAL       | Pin A8, A9, A10, A11, A12, A13 (Arduino Mega) |
| Giroscopio  | Conexiones                                    |
| VCC         | 5v (Arduino Mega)                             |
| GND         | GND (Arduino Mega)                            |
| SDA         | Pin 20 (Arduino Mega SDA)                     |
| SCL         | Pin 21 (Arduino Mega SCL)                     |

Fuente. Autor

### 3. Programación del código del guante háptico

Para la programación se distribuye en cuatro partes que son el código de los sensores Flex, el giroscopio, los servomotores, y los módulos motor-vibradores para ello se explicará el funcionamiento del código.

#### a) Programación de los sensores Flex

Para la programación de los sensores Flex se utilizó una variable extra para cada sensor con el fin de almacenar los datos individuales, utilizando analogRead se tomó lecturas de medición.

```
flex1Value = analogRead(flex1);
flex2Value = analogRead(flex2);
flex3Value = analogRead(flex3);
flex4Value = analogRead(flex4);
flex5Value = analogRead(flex5);
flex6Value = analogRead(flex6);
```

#### b) Programación del giroscopio

Para la programación del giroscopio se utilizó las librerías “MPU6050, SPI, I2Cdev y Wire” con el fin de tener una adquisición de datos más precisos, sin embargo, el sensor giroscopio era muy sensible a vibraciones por lo que se utilizó una programación para filtrar los datos y obtener mejores resultados.

```
// Leer las aceleraciones y velocidades angulares
sensor.getAcceleration(&ax, &ay, &az);

//Calcular los ángulos de inclinación:
float accel_ang_x = atan(ax/sqrt(pow(ay,2) + pow(az,2))) * (180.0/3.14);
float accel_ang_y = atan(ay/sqrt(pow(ax,2) + pow(az,2))) * (180.0/3.14);

//*****FILTROS EJE "X" *****//
// restamos la última lectura
```

```

total = total - readings[index];
// leemos del sensor
readings[index] = accel_ang_x;
// añadimos la lectura al total
total = total + readings[index];
// avanzamos a a próxima posición del array
index = index + 1;

// si estamos en el final del array
if (index >= numReadings)
    index = 0;

// calculo del promedio
averageX = total / numReadings;
// lo mandamos a la PC como valor ASCII

//*****FILTROS EJE "Y" *****//
// restamos la última lectura
total2 = total2 - readings2[index2];
// leemos del sensor
readings2[index2] = accel_ang_y;
// añadimos la lectura al total
total2 = total2 + readings2[index2];
// avanzamos a próxima posición del array
index2 = index2 + 1;

// si estamos en el final del array
if (index2 >= numReadings2)
    index2 = 0;

// calculo del promedio
averageY = total2 / numReadings2;
// lo mandamos a la PC como valor ASCII

X1 = averageX;
Y1 = averageY;

```

Realizando estos filtros se mejora la señal Permitiendo que el puente háptico responda de mejor forma y evitando los picos en las mediciones como se muestra en la Figura 5.

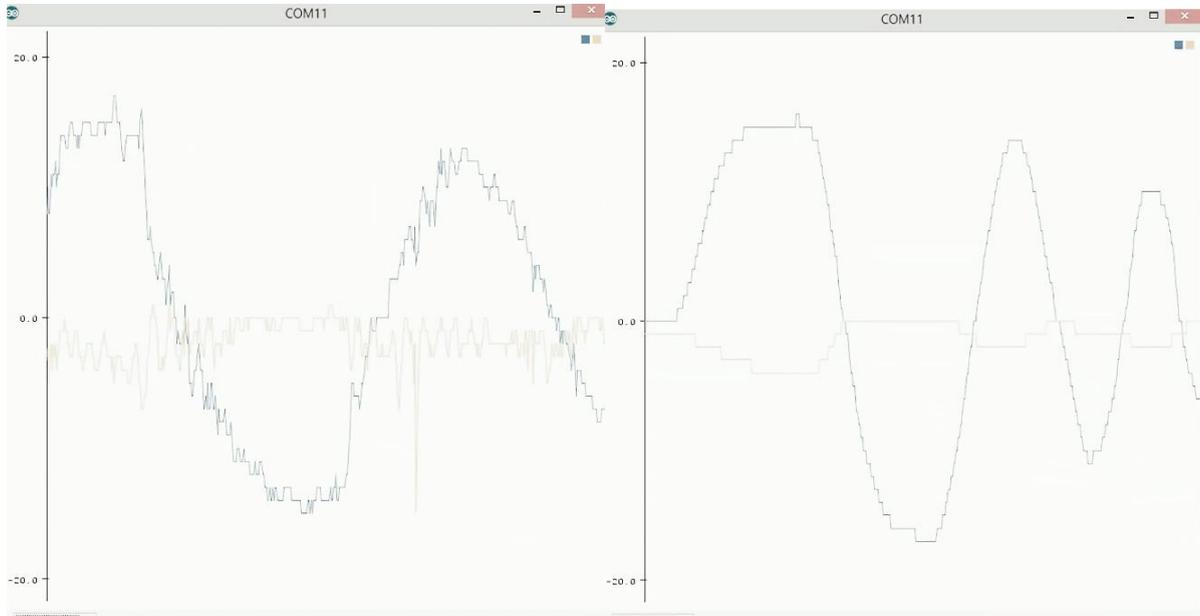


Figura. 68 Mediciones del giroscopio sin filtros y con filtros

Fuente. Autor

Como se observa en la Figura 5 la imagen de la izquierda son las mediciones del giroscopio sin filtros en la que se identifica muchas perturbaciones por las vibraciones con movimientos que se generan en el giroscopio mientras que en la imagen de la derecha se observa una gráfica sin perturbaciones y esto permite una estabilidad para el guante háptico.

### c) Programación de los servomotores

Para programar los servomotores y utilizó la librería "ServoTimer2", ya que con la librería normal del servo no se puede trabajar debido a la librería de comunicación el módulo LoRa. Esta librería funciona en valores de 750 a 2250 siendo 750 equivalente a 0° y 2250 a 180° para ello se declaró condiciones en las cuales por medio del movimiento del giroscopio y las señales que envían los sensores del vehículo al exoesqueleto del guante háptico se abra o se cierre.

```
(Y2 >= 19 || Y1 >= 19) {
  if (Y2 >= 19 && Y2 <= 26 || Y1 >= 19 && Y1 <= 26) { // BAJO
    ServoMotor1.write(2000); // bajar de 2250 a 2000
```

```

    ServoMotor2.write(2250); // bajar de 2250 a 2000
    ServoMotor3.write(2250); // bajar de 2250 a 2000
    ServoMotor4.write(950); // subir de 750 a 900
    ServoMotor5.write(1100); // subir de 750 a 900
}
if (Y2 >= 26 && Y2 <= 33 || Y1 >= 26 && Y1 <= 33) { // MEDIO
    ServoMotor1.write(1850); // bajar de 2250 a 2000
    ServoMotor2.write(1900); // bajar de 2250 a 2000
    ServoMotor3.write(1900); // bajar de 2250 a 2000
    ServoMotor4.write(1300); // subir de 750 a 900
    ServoMotor5.write(1500); // subir de 750 a 900
}
if (Y2 >= 40 || Y1 >= 40) { // ALTO
    ServoMotor1.write(1650); // bajar de 2250 a 2000
    ServoMotor2.write(1750); // bajar de 2250 a 2000
    ServoMotor3.write(1600); // bajar de 2250 a 2000
    ServoMotor4.write(1450); // subir de 750 a 900
    ServoMotor5.write(1650); // subir de 750 a 900
}

```

#### d) Programación de los motores vibradores

Para programar los motores vibradores se declaró condiciones en las cuales por medio de las señales que envían los sensores del vehículo al exoesqueleto del guante háptico vibra en las puntas de los dedos se encuentran situados los vibradores con el fin de que cada sensor del vehículo acciones un motor vibrador, para el motor vibrador del pulgar se lo declaró que vibre acorde a la velocidad del vehículo a mayor velocidad mayor vibración y viceversa.

```

// INDICE DISTANCIA IZQUIERDA
if (d3 <= 8) {
digitalWrite(vibrador2, 1);
} else {
digitalWrite(vibrador2, 0);
}

// MEDIO DISTANCIA FRENTE
if (d2 <= 8) {
digitalWrite(vibrador3, 1);
} else {
digitalWrite(vibrador3, 0);
}

// ANULAR DISTANCIA DERECHA
if (d1 <= 8) {
digitalWrite(vibrador4, 1);
} else {
digitalWrite(vibrador4, 0);
}

// MEÑIQUE DISTANCIA ATRAS
if (d4 <= 8) {
digitalWrite(vibrador5, 1);
} else {
digitalWrite(vibrador5, 0);}

```

#### 4. Programación del código del vehículo

Para la programación se distribuye en dos partes que son el código de los sensores Sharp, y los motores Pololu para ello se explicará el funcionamiento del código.

##### a) Programación de los sensores Sharp

Para realizar la programación de los sensores Sharp, se realizó 10 lecturas de los datos tomados del sensor de distancia para sacar un promedio, con el fin de reducir los picos que se producían al tomar las lecturas y trabajar con un valor más preciso.

```
D_cm1 = distancia1(10); //lectura de distancia
D_cm2 = distancia2(10); //lectura de distancia
D_cm3 = distancia3(10); //lectura de distancia
D_cm4 = distancia4(10); //lectura de distancia

// MEDICION DE DISTANCIAS
float distancia1(int n1) {
    long suma1 = 0;
    for (int i = 0; i < n1; i++) {
        suma1 = suma1 + analogRead(sensor1);}
    float adc1 = suma1 / n1;
    float distancia_cm1 = 17569.7 * pow(adc1, -1.2062);
    return (distancia_cm1);}
```

##### b) Programación de los motores Pololu y servo motor

El código para controlar los motores pololu se comunica con el guante háptico ya que los sensores Flex son los encargados de enviar la señal para controlar la velocidad y el giroscopio controla la dirección de giro del servomotor al igual que la marcha y reversa de los motores Pololu

```
// CONTROL DE LA DIRECCION SERVOMOTOR
if (angulo >= 1150 && angulo <= 1375) {
    ServoMotor.write(1275);
```

```

    } else {
        ServoMotor.write(angulo);
    }

// CONTROL DE VELOCIDAD DE LOS MOTORES POLOLU
void mover(int motor, int velocidad, int direccion) {
    //Vamos a definir la función mover, que va a acciona un motor, fijar
    su velocidad y el sentido de giro. Definimos:
    //motor: llamaremos 1 al motor A, y 2 al motor B
    //velocidad: desde 0 a 255
    //dirección: 0 para el giro en sentido horario (delante), 1 sentido
    antihorario (atras)
    digitalWrite(STBY, HIGH);           //deshabilitar standby para mover
    boolean inPin1 = LOW;               // creamos la variable booleana (solo
    puede ser HIGH/LOW) inpin1
    boolean inPin2 = HIGH;             // y le asignamos el valor LOW. A
    inPin2 le damos el valor          HIGH
    if (direccion == 1) {
        inPin1 = HIGH;
        inPin2 = LOW;
    }
    if (motor == 1) {
        digitalWrite(AIN1, inPin1);
        digitalWrite(AIN2, inPin2);
        analogWrite(PWMA, velocidad);
    } else {
        digitalWrite(BIN1, inPin1);
        digitalWrite(BIN2, inPin2);
        analogWrite(PWMB, velocidad);
    }
}

void stop() { //ahora si, accionamos el standby
    digitalWrite(STBY, LOW);
}

```

## 5. Comunicación Shield LoRa

La comunicación lora es bidireccional por lo que el vehículo y el guante háptico envía y recibe información Para ello se utiliza las librerías “SPI, RH\_RF95”, lo que permite enviar y recibir información en conjunto y luego de ello procesarlas para utilizarlas individualmente.

```

String data = String(D_cm1) + "-" + String(D_cm2) + "-" + String(D_cm3) + "-"
" + String(D_cm4);
int dataLength = data.length(); dataLength++;
uint8_t total[dataLength];
//Serial.println(data);
data.toCharArray(total, dataLength);
rf95.send(total, dataLength);
rf95.waitPacketSent();

uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);

if (rf95.waitForAvailable(60)) {
  //if (rf95.available()) {
  if (rf95.recv(buf, &len)) {
    String dataTotal = (char*)buf;
    explode(dataTotal);
  }
  else {
    Serial.println("RECEIVED FAILED");
  }
}
else {
  Serial.println("no hay datos de entrada");
}

//VARIABLES RECIBIDAS
void explode(String req) {
  char str[30];
  req.toCharArray(str, 20);
  char * pch;
  pch = strtok (str, ",");
  while (pch != NULL) {
    turn++;
    if (turn == 1) {
      Serial.print("IZQ DER X: ");
      X = atof(pch);
      Serial.print(X);
    }
    if (turn == 2) {
      Serial.print(" ARR BAJ Y: ");
      Y = atof(pch);
      Serial.print(Y);
    }
    if (turn == 3) {
      Serial.print(" VEL FL: ");
      V = atof(pch);
      Serial.print(V);
    }
    if (turn == 4) {
      Serial.print(" PITO FL: ");
      P = atof(pch);
      Serial.print(P);
    }
    pch = strtok (NULL, ",");
    //delay(100);
  }
  turn = 0;
}

```

## 6. Aspectos a tener en cuenta

- Antes de conectar la batería se debe asegurar que el interruptor este apagado (0), tanto del vehículo como del guante háptico.
- Antes de encender el vehículo verificar que las antenas de los Shield LoRa se encuentren bien sujetos ya que esto ocasiona distorsión e interrupción de la comunicación entre el guante háptico y el vehículo.
- Antes de colocar los dedos en el exoesqueleto encender ambos dispositivos y verificar el emparejamiento ya que al inicio antes de emparejarse se ejerce un fuerza de los servomotores.
- Verificar las conexiones del exoesqueleto con la base del ante brazo acorde a la numeración especificada ya que si se conecta en los pines equivocados podría sufrir fallas o un mal funcionamiento.
- Cuando se encienda el prototipo y se observe un mal funcionamiento se debe apagar y encender con el fin de que se establezca correctamente la comunicación de los dispositivos Shield LoRa.
- Se recomienda no forzar el funcionamiento del guante háptico y del vehículo cuando se note una deficiencia de rendimiento ya que se debe al desgaste de energía de la batería y necesita ser cargada ya que un mal uso acabaría con la vida útil de la batería.
- Para cargar la batería Li-Po se debe conectar los cables más delgados que vienen con un conector macho blanco a un cargador de baterías Li-Po. Al momento de cargar

las baterías verificar el voltaje total y de cada celda. Por lo general el voltaje de carga máximo es 8.4v para el guante háptico y 8.2v del vehículo, pero no es recomendable cargarle hasta ese valor por lo que se recomienda cargar la batería hasta un máximo de 8v a 7.8v.

## **7. Mejoras futuras**

- El vehículo cuenta con una placa distribuidora de voltaje la cual tiene salidas de voltaje de 12v y 5v que aún pueden ser utilizadas para poder conectar algún otro dispositivo electrónico.
- El controlador Arduino Mega dispone de más salidas digitales y analógicas si se quiere añadir algún tipo de sensor u otro componente.
- El uso de antenas de señal mejora la comunicación y transmisión de datos ya que brinda un alcance a distancias más largas por lo que se podría adaptar mejores tipos de antenas si se requiere establecer comunicaciones a largas distancias.
- Se puede utilizar componentes más pequeños para disminuir el volumen y tamaño del guante háptico, pero esto conlleva a costos un poco más elevados.

## **Anexo J: Manual de Usuario**

### **INTRODUCCIÓN**

La implementación de cualquier tipo de prototipo conlleva la realización de un manual de usuario con el fin de detallar toda la configuración de los dispositivos utilizados para que cualquier usuario que tenga acceso al prototipo tenga la facilidad de controlarlo y conozca cuál es el funcionamiento del prototipo al igual que su modo de ejecución. Para ello se detalla desde la configuración inicial que se realizó, hasta la ejecución y funcionamiento del prototipo.

#### **1. Colocación del guante háptico**

La colocación del guante háptico consta de cuatro partes las cuales son la colocación del guante elástico, de la caja de control que se ubica en el antebrazo, del exoesqueleto y las conexiones de la caja de control con el exoesqueleto.

##### **a) Colocación del guante elástico**

Primero se inicia colocando el guante elástico en la mano derecha del usuario sujetando bien con una cinta adhesiva para poder maniobrar correctamente a la hora de utilizar el prototipo como se muestra en la Figura 1.

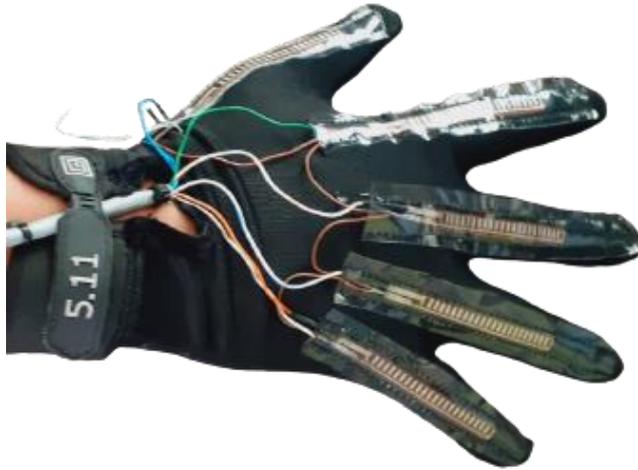


Figura. 69 Colocación del guante elástico

Fuente. Autor

### b) Colocación de la caja de control

Segundo se coloca la caja de control en la parte del antebrazo ubicada desde el inicio de la muñeca con el fin de poder conectar los cables y evitando que se genere un estiramiento de los cables. Se debe sujetar bien con las 2 correas adhesivas para que no exista Movimiento de irregulares como se observa en la Figura 2.



Figura. 70 Sujeción de la caja de control

Fuente. Autor

### c) Colocación del exoesqueleto

Como tercer punto se procede a colocarse el exoesqueleto en la parte de la muñeca sujetando fuertemente con la correa adhesiva. una vez realizado este proceso se pasa la colocación de los dedos en cada compartimiento del exoesqueleto como se observa en la Figura 3.



Figura. 71 Sujeción del exoesqueleto

Fuente. Autor

#### d) Conexiones de la caja de control con el exoesqueleto

Para finalizar una vez sujeto tanto la caja de control como el exoesqueleto se procede a realizar las conexiones del cuadro de control con el exoesqueleto colocando en los pines acordé a la numeración de cada cable como se observa en la Figura 4.



Figura. 72 Conexiones de los pines del exoesqueleto y la caja de control

Fuente. Autor

## 2. Funcionamiento del prototipo

Para que el guante háptico empiece su funcionamiento se debe verificar que los interruptores este apagados (OFF), para luego proceder a conectar la batería del vehículo y del guante háptico como se observa en la Figura 5.

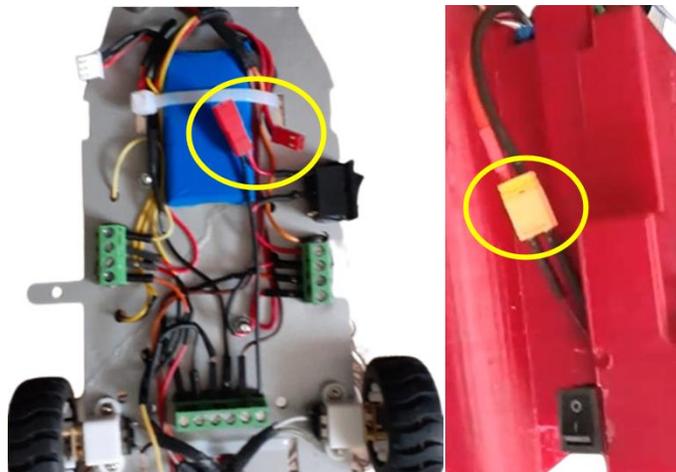


Figura. 73 Sujeción de la caja de control

Fuente. Autor

Antes de colocar los dedos en el exoesqueleto encender primero los dispositivos ya que inicialmente cuando se comunican el guante háptico tiende a mover el exoesqueleto.

Luego se procede a encender el vehículo desde su interruptor al igual que el guante háptico, el cual debe estar en la posición OFF (O) para cambiarla a ON (I) como se observa en la Figura 6. Una vez realizado este paso el guante óptico empezará a funcionar, en el caso que exista un mal funcionamiento apagar y encender nuevamente los interruptores, esto se debe a la falla de comunicación de los dispositivos LoRa.



Figura. 74 Encendido del vehículo y el guante háptico

Fuente. Autor

Una vez enlazada las comunicaciones entre el vehículo y el guante háptico ya se puede utilizar el prototipo de la siguiente forma:

- Moviendo el brazo hacia arriba damos la dirección al vehículo para el retro.
- Moviendo el brazo hacia abajo damos la dirección al vehículo para la marcha.
- El movimiento del brazo a la izquierda da la dirección del vehículo en ese sentido.
- El movimiento del brazo a la derecha da la dirección del vehículo en ese sentido.
- Al momento de levantar la mano el giroscopio actúa y da la indicación para que el guante háptico se cierre y envíe esa señal al vehículo para que acelere en dirección reversa acorde a cómo se cierre o flexione el guante háptico.
- De igual manera al bajar la mano el giroscopio actúa y da la indicación para que el guante a tico se cierre y envíe la señal del vehículo para que acelere en dirección hacia adelante acorde a cómo se cierre o flexione el guante háptico.

- El vehículo tiene cuatro sensores de distancia los cuales al momento de detectar un obstáculo envían una señal al guante háptico, indicando que el vehículo no puede continuar su recorrido porque tiene un bloqueo y la punta de un dedo del guante háptico empieza a vibrar para que el usuario deje de acelerar y realice otra acción como por ejemplo girar o dar reversa.
- Cada sensor de distancia del vehículo controla un dedo del guante háptico para que vibre e indique al usuario la dirección del obstáculo que detecta el vehículo.
- La velocidad del vehículo está controlada por el movimiento del puente público la cual el dedo pulgar del guante vibra acorde a la velocidad es decir a mayor velocidad más vibración y a menor velocidad menos vibración.
- En posición horizontal del brazo, se flexiona el dedo meñique y se enciende led 1 delantero derecho.
- En posición horizontal del brazo, se flexiona el dedo anular y se enciende led 2 delantero izquierdo.
- En posición horizontal del brazo, se flexiona el dedo medio y se enciende led 3 trasero derecho.
- En posición horizontal del brazo, se flexiona el dedo índice y se enciende led 4 trasero izquierda.
- En posición horizontal del brazo se flexiona el dedo pulgar y se enciende todos los leds parpadeando.

### **3. Cargar programación**

Cuando exista algún problema en el funcionamiento del prototipo es recomendable volver a cargar las programaciones de cada dispositivo para ello se presenta los pasos a seguir para que todo funcione correctamente.

1. Antes de cargar una programación se debe verificar las librerías que se usan para ejecutar el código correctamente como se muestra en la Figura 7.

```
// Librerías I2C para controlar el mpu6050
// la librería MPU6050.h necesita I2Cdev.h, I2Cdev.h necesita Wire.h
#include <ServoTimer2.h>
#include <SPI.h>
#include <RH_RF95.h>
#include <MPU6050.h>
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"
```

Figura. 75 Verificación de librerías

Fuente. Autor

En el caso de no disponer de estas librerías se debe realizar estos pasos.

- a) Cargar directamente del buscador de Arduino como se muestra en la Figura 8.

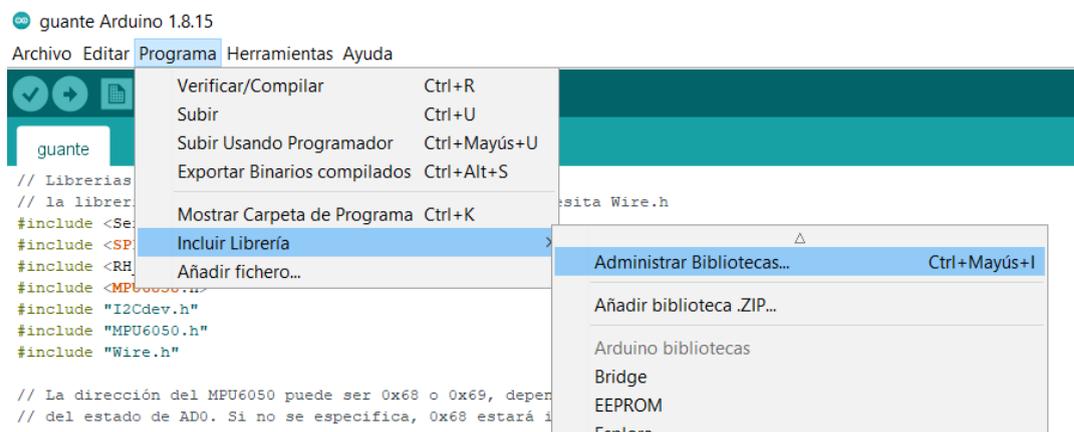


Figura. 76 Verificación de librerías

Fuente. Autor

- b) Buscar la librería que no se disponga e instalar como se muestra en la Figura 9.

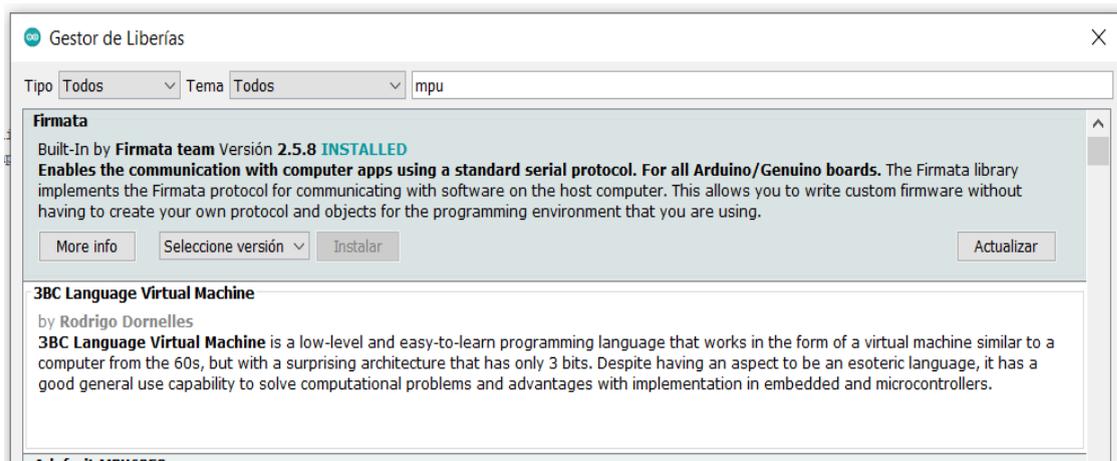


Figura. 77 Buscador de librerías

Fuente. Autor

- Una vez instaladas las librerías necesarias se procede a verificar la programación para verificar que todo este correcto como se muestra en la Figura 10.

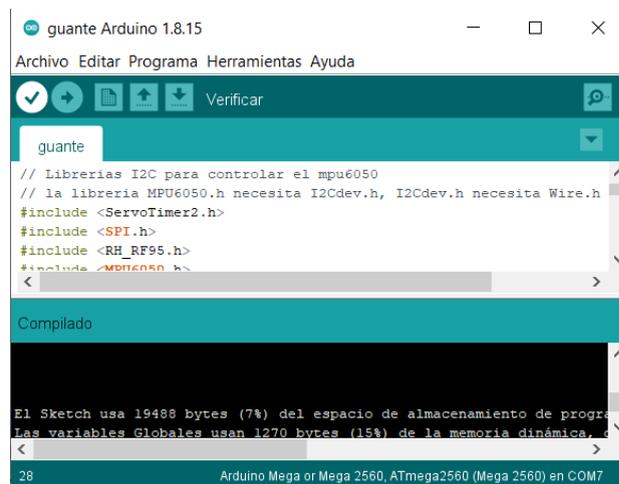


Figura. 78 Verificación y compilación de la programación

Fuente. Autor

- Para cargar un programa en Arduino, debemos seleccionar siempre el modelo de la placa conectada y el puerto al que está conectado como se muestra en la Figura 11.

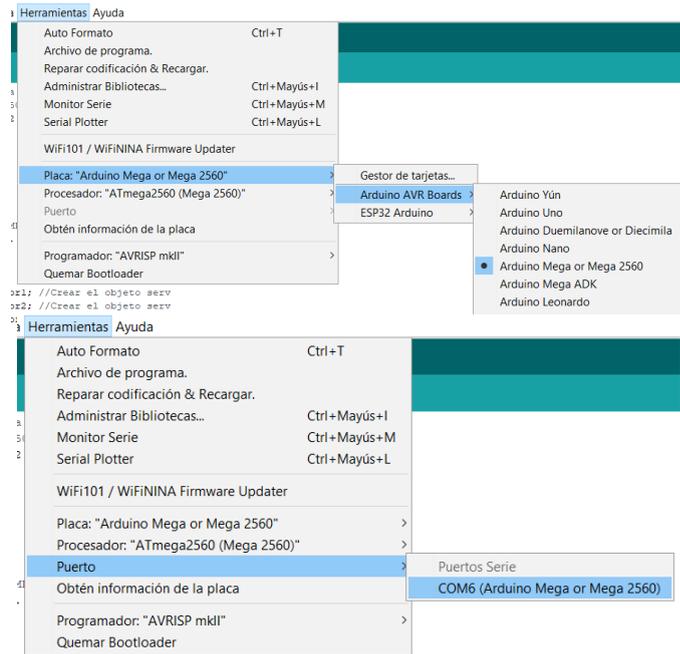


Figura. 79 Selección de placa y puerto del Arduino

Fuente. Autor

4. Por último, damos en subir programa y verificamos que se cargué correctamente el código en el Arduino como se muestra en la Figura 12.



Figura. 80 Subir programa al Arduino Mega

Fuente. Autor

#### 4. Problemas y soluciones

- Cuando los módulos Shield LoRa no se comunican entre sí se debe apagar y encender el vehículo y el guante háptico para que se reinicie el emparejamiento entre ellos.
- Si el prototipo sobrepasa la obra de funcionamiento y se empieza a notar una deficiencia en el rendimiento o un mal funcionamiento de las condiciones declaradas, se debe detener las pruebas y proceder a cargar las baterías del guante háptico y del vehículo.
- Antes de realizar las pruebas verificar que los sensores Sharp de distancia se encuentre completamente limpio en la parte frontal de su lente ya que algún residuo podría ocasionar datos erróneos y un mal funcionamiento del prototipo.
- Sujetar bien las correas de la caja de control en el brazo y del exoesqueleto en la mano ya que esto puede causar movimientos no deseados que interfieran en el correcto funcionamiento del prototipo.
- Cuando el funcionamiento del prototipo es incorrecto verificar que las antenas de los dispositivos Shield LoRa se encuentren bien sujetas ya que esto causa que no se estén comunicando correctamente el guante háptico y el vehículo.

## **Anexo K: Manual de mantenimiento**

### **INTRODUCCIÓN**

La implementación de cualquier tipo de prototipo conlleva la realización de un manual de mantenimiento con el fin de detallar las revisiones que se debe realizar para que cualquier usuario que pueda acceder al prototipo tenga en cuenta el mantenimiento que se deba realizar para alargar la vida útil del prototipo.

#### **1. Mantenimiento para realizarse antes de su uso**

Cada vez que el prototipo vaya a ser usado se debe verificar algunos aspectos para no tener problemas a futuro como los que se menciona a continuación.

- Antes de conectar la batería al vehículo, revisar que los interruptores de encendido y apagado de la alimentación se encuentren apagados tanto del guante háptico como del vehículo.
- Revisar si existe impurezas en los ejes de las llantas para evitar acumulaciones de basura y desgaste en el eje de la llanta.
- Verificar que la antena del transmisor y receptor de datos de los módulos Shield LoRa este bien colocados y sujetos, para evitar daños en el equipo o que su funcionamiento no sea eficiente.
- Verificar si hay una deficiencia en el rendimiento del prototipo para cargar las baterías y no exceder el tiempo de uso de 1 hora para evitar daños futuros y alargar la vida útil de la batería.
- Revisar que no exista impurezas en los sensores Sharp de distancia ya que podría provocar una mala detección de obstáculos y no funcionar correctamente.

- Al finalizar el uso del prototipo primero apagar los interruptores del guante háptico y del vehículo y luego desconectar el conector de la batería para evitar alguna fuga de voltaje cuando no esté en funcionamiento.
- No colocar encima del guante elástico objetos pesados ya que esto podría ocasionar problemas con los sensores Flex.

## **2. Mantenimiento preventivo**

Este tipo de mantenimiento se lo debe realizar dependiendo el uso del guante háptico, si su uso es muy frecuente lo más aconsejable es cada 15 días, si el uso es frecuente cada mes, y si es poco frecuente cada dos meses. Los aspectos para revisar son los siguientes:

- Verificar si los tornillos estén bien sujetos o exista una alguna parte móvil realizar un ajuste.
- Verificar que los motores del vehículo no tengan impurezas caso contrario limpiar las obstrucciones.
- Realizar un engrase al eje del motor para que no exista ningún rose o calentamiento.
- Verificar que no existe impurezas en el servomotor para que pueda girar libremente el vehículo, de igual forma los servomotores del exoesqueleto así evitando desgaste del eje de los servomotores.
- Verificar continuidad en los conductores.
- Limpiar las impurezas como polvos en la placa de distribución de voltaje y todos los componentes electrónicos.
- Verificar voltajes de operación tanto del Arduino como de la placa distribuidora de voltaje del guante háptico y del vehículo.
- Verificar que las piezas del exoesqueleto se encuentren en buen estado caso contrario para el uso y remplazar imprimiendo en 3D la pieza defectuosa.

## Anexo L: Programación del guante háptico

```
// Librerías I2C para controlar el mpu6050
// la librería MPU6050.h necesita I2Cdev.h, I2Cdev.h necesita Wire.h
#include <ServoTimer2.h>
#include <SPI.h>
#include <RH_RF95.h>
#include <MPU6050.h>
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

// La dirección del MPU6050 puede ser 0x68 o 0x69, dependiendo
// del estado de AD0. Si no se especifica, 0x68 estará implícito
MPU6050 sensor;
RH_RF95 rf95;

ServoTimer2 ServoMotor1; //Crear el objeto serv
ServoTimer2 ServoMotor2; //Crear el objeto serv
ServoTimer2 ServoMotor3; //Crear el objeto serv
ServoTimer2 ServoMotor4; //Crear el objeto serv
ServoTimer2 ServoMotor5; //Crear el objeto serv

// pines de los servomotores
int servo1 = 8; // Declaramos pin del servo pulgar
int servo2 = 13; // Declaramos pin del servo
int servo3 = 12; // Declaramos pin del servo
int servo4 = 11; // Declaramos pin del servo
int servo5 = 7; // Declaramos pin del servo meñique

// pines vibradores
int vibrador1 = 10;
int vibrador2 = 6;
int vibrador3 = 5;
int vibrador4 = 4;
int vibrador5 = 3;
int motorvibrador;

// pines del sensor flex
int flex1 = A8;
int flex2 = A9;
int flex3 = A10;
int flex4 = A11;
int flex5 = A12;
int flex6 = A13;

// Variables a sensores flex
int flex1Value;
int flex2Value;
int flex3Value;
int flex4Value;
int flex5Value;
int flex6Value;
int flexTotal;

// variables de giroscopio en eje X y Y
int X1;
int Y1;
int Y2;
// variables mapeo exoesqueleto
```

```

//int angulo1;
//int angulo2;

int d1;
int d2;
int d3;
int d4;

int turn = 0;

// Valores RAW (sin procesar) del acelerometro y giroscopio en los ejes x,y,z
int ax, ay, az;
//int gx, gy, gz;

//***** FILTRO X *****//
const int numReadings = 40; // numero de muestras para promedio
int readings[numReadings]; // lecturas de la entrada analogica
int index = 0; // el indice de la lectura actual
int total = 0; // total
float averageX = 0; // promedio

//***** FILTRO Y *****//
const int numReadings2 = 40; // numero de muestras para promedio
int readings2[numReadings2]; // lecturas de la entrada analogica
int index2 = 0; // el indice de la lectura actual
int total2 = 0; // total
float averageY = 0; // promedio

void setup() {
  pinMode(vibrador1, OUTPUT);
  pinMode(vibrador2, OUTPUT);
  pinMode(vibrador3, OUTPUT);
  pinMode(vibrador4, OUTPUT);
  pinMode(vibrador5, OUTPUT);

  ServoMotor1.attach(serv01); // Declarar el servomotor al pin
  ServoMotor2.attach(serv02); // Declarar el servomotor al pin
  ServoMotor3.attach(serv03); // Declarar el servomotor al pin
  ServoMotor4.attach(serv04); // Declarar el servomotor al pin
  ServoMotor5.attach(serv05); // Declarar el servomotor al pin

  Serial.begin(115200); //Iniciando puerto serial

  for (int thisReading = 0; thisReading < numReadings; thisReading++)
    readings[thisReading] = 0;
  for (int thisReading2 = 0; thisReading2 < numReadings2; thisReading2++)
    readings2[thisReading2] = 0;

  Wire.begin(); //Iniciando I2C
  sensor.initialize(); //Iniciando el sensor

  if (sensor.testConnection()) Serial.println("Sensor iniciado correctamente");
  else Serial.println("Error al iniciar el sensor");

  while (!Serial) ; // Wait for serial port to be available
  if (!rf95.init())
    Serial.println("Fallo inicio");
  //delay(2000);
}

```

```

void loop() {
  // Leer las aceleraciones y velocidades angulares
  sensor.getAcceleration(&ax, &ay, &az);

  //Calcular los angulos de inclinacion:
  float accel_ang_x = atan(ax / sqrt(pow(ay, 2) + pow(az, 2))) * (180.0 /
3.14);
  float accel_ang_y = atan(ay / sqrt(pow(ax, 2) + pow(az, 2))) * (180.0 /
3.14);
  //*****FILTROS EJE "X" *****//
  // restamos la ultima lectura
  total = total - readings[index];
  // leemos del sensor
  readings[index] = accel_ang_x;
  // añadimos la lectura al total
  total = total + readings[index];
  // avanzamos a a proxima posiscion del array
  index = index + 1;

  // si estamos en el final del array
  if (index >= numReadings)
    index = 0;

  // calculamos el promedio
  averageX = total / numReadings;
  // lo mandamos a la PC como valor ASCII

  //*****FILTROS EJE "Y" *****//
  // restamos la ultima lectura
  total2 = total2 - readings2[index2];
  // leemos del sensor
  readings2[index2] = accel_ang_y;
  // añadimos la lectura al total
  total2 = total2 + readings2[index2];
  // avanzamos a a proxima posiscion del array
  index2 = index2 + 1;

  // si estamos en el final del array
  if (index2 >= numReadings2)
    index2 = 0;

  // calculamos el promedio
  averageY = total2 / numReadings2;
  // lo mandamos a la PC como valor ASCII

  X1 = averageX;
  Y1 = averageY;
  //Serial.print("  Y1: ");
  //Serial.println(Y1);

  flex1Value = analogRead(flex1);
  flex2Value = analogRead(flex2);
  flex3Value = analogRead(flex3);
  flex4Value = analogRead(flex4);
  flex5Value = analogRead(flex5);
  flex6Value = analogRead(flex6);

  flexTotal = (flex1Value + flex2Value + flex3Value + flex4Value +
flex5Value) / 5;

```

```

motorvibrador = map(flexTotal, 235, 380, 255, 0);
/*Serial.print("sensor 1: ");
  Serial.print(flex1Value);
  Serial.print("  sensor 2: ");
  Serial.print(flex2Value);
  Serial.print("  sensor 3: ");
  Serial.print(flex3Value);
  Serial.print("  sensor 4: ");
  Serial.print(flex4Value);
  Serial.print("  sensor 5: ");
  Serial.println(flex5Value);*/
//delay(20);
//Serial.print("  flex: ");
//  Serial.print(flexTotal);

String data = String(X1) + "," + String(Y1) + "," + String(flexTotal) +
"," + String(flex6Value)+ "," + String(flex1Value)+ "," + String(flex2Value)+
"," + String(flex3Value)+ "," + String(flex4Value)+ "," + String(flex5Value);
int dataLength = data.length(); dataLength++;
uint8_t total[dataLength];
data.toCharArray(total, dataLength);
//Serial.println(data);

if (rf95.waitForAvailable(1)) {
  if (rf95.available()) {
    // Now wait for a reply
    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);

    if (rf95.recv(buf, &len)) {
      String dataTotal = (char*)buf;
      explode(dataTotal);

      rf95.send(total, dataLength);
      rf95.waitForPacketSent();
    }
  } else {
    Serial.println("fallo recibido");
  }
}
//delay(400);
// ***** CONDICIONES DE MANDO *****

if (Y1 <= 0) {
  Y2 = Y1 * -1;
}
//Serial.print("  Y2: ");
//Serial.print(Y2);

if (Y2 >= 19 || Y1 >= 19) {
  if (Y2 >= 19 && Y2 <= 26 || Y1 >= 19 && Y1 <= 26) { // BAJO
    ServoMotor1.write(2000); // bajar de 2250 a 2000
    ServoMotor2.write(2250); // bajar de 2250 a 2000
    ServoMotor3.write(2250); // bajar de 2250 a 2000
    ServoMotor4.write(950); // subir de 750 a 900
    ServoMotor5.write(1100); // subir de 750 a 900
  }
  if (Y2 >= 26 && Y2 <= 33 || Y1 >= 26 && Y1 <= 33) { // MEDIO
    ServoMotor1.write(1850); // bajar de 2250 a 2000
    ServoMotor2.write(1900); // bajar de 2250 a 2000
  }
}

```

```

        ServoMotor3.write(1900); // bajar de 2250 a 2000
        ServoMotor4.write(1300); // subir de 750 a 900
        ServoMotor5.write(1500); // subir de 750 a 900
    }
    if (Y2 >= 40 || Y1 >= 40) { // ALTO
        ServoMotor1.write(1650); // bajar de 2250 a 2000
        ServoMotor2.write(1750); // bajar de 2250 a 2000
        ServoMotor3.write(1600); // bajar de 2250 a 2000
        ServoMotor4.write(1450); // subir de 750 a 900
        ServoMotor5.write(1650); // subir de 750 a 900
    }
}

// PULGAR VIBRACION CON VELOCIDAD
if (motorvibrador <= 50) {
    digitalWrite(vibrador1, 0);
} else {
    analogWrite(vibrador1, motorvibrador);
}
int obst = 13;
// OBSTACULO DE LOS SENSORES SHARP
if (d1 <= obst || d2 <= obst || d3 <= obst || d4 <= obst) {
    ServoMotor1.write(2100); // bajar de 1700 a 1700
    ServoMotor2.write(2250); // bajar de 2250 a 1700
    ServoMotor3.write(2250); // bajar de 2250 a 1700
    ServoMotor4.write(850); // subir de 750 a 1800
    ServoMotor5.write(1000); // subir de 900 a 1500
}

// INDICE DISTANCIA IZQUIERDA
if (d3 <= obst) {
    digitalWrite(vibrador2, 1);
} else {
    digitalWrite(vibrador2, 0);
}

// MEDIO DISTANCIA FRENTE
if (d2 <= obst) {
    digitalWrite(vibrador3, 1);
} else {
    digitalWrite(vibrador3, 0);
}

// ANULAR DISTANCIA DERECHA
if (d1 <= obst) {
    digitalWrite(vibrador4, 1);
} else {
    digitalWrite(vibrador4, 0);
}

// MEÑIQUE DISTANCIA ATRAS
if (d4 <= obst) {
    digitalWrite(vibrador5, 1);
} else {
    digitalWrite(vibrador5, 0);
}
}

```

```

void explode(String req) {
  char str[40];
  req.toCharArray(str, 20);
  char * pch;
  pch = strtok (str, "-");
  while (pch != NULL) {
    //String sementara = pch;
    turn++;
    if (turn == 1) {
      Serial.print("Dist 1: ");
      d1 = atof(pch);
      Serial.print(d1);
    }
    if (turn == 2) {
      Serial.print("      Dist 2: ");
      d2 = atof(pch);
      Serial.print(d2);
    }
    if (turn == 3) {
      Serial.print("      Dist 3: ");
      d3 = atof(pch);
      Serial.print(d3);
    }
    if (turn == 4) {
      Serial.print("      Dist 4: ");
      d4 = atof(pch);
      Serial.println(d4);
    }
    pch = strtok (NULL, "-");
    //delay(100);
  }
  turn = 0;
}

```

## Anexo M: Programación del vehículo

```
#include <ServoTimer2.h>
#include <SPI.h>
#include <RH_RF95.h>

RH_RF95 rf95;
ServoTimer2 ServoMotor; //Crear el objeto serv
int servo = 11; // Declaramos pin del servo
int X;
int Y;
int V;
int P;
int F1; int F2; int F3; int F4; int F5;
int start = 0;

int angulo;

//para el motor A
int PWMA = 3; // control de velocidad
int AIN2 = 4; // dirección
int AIN1 = 5; // dirección
int STBY = 6; // definimos el pin de standby
// para el motor B
int BIN1 = 7; // dirección
int BIN2 = 8; // dirección
int PWMB = 12; // control de velocidad*/
int estado;
int vel;

// PINES A USAR DE LOS SENSORES
int sensor1 = A1;
int sensor2 = A2;
int sensor3 = A3;
int sensor4 = A4;
// VARIABLES ED LOS SENSORES
int D_cm1;
int D_cm2;
int D_cm3;
int D_cm4;

// LUCES
int led_der_d = 23;
int led_izq_d = 35;
int led_der_a = 27;
int led_izq_a = 41;

//buzzer
int Buzzer = 46;
byte salida = 220;
int turn = 0;

void setup() {
  Serial.begin(115200);
  pinMode(STBY, OUTPUT);
  pinMode(PWMA, OUTPUT);
  pinMode(AIN1, OUTPUT);
  pinMode(AIN2, OUTPUT);
  pinMode(PWMB, OUTPUT);
  pinMode(BIN1, OUTPUT);
```

```

pinMode(BIN2, OUTPUT);
ServoMotor.attach(servo); // Declarar el servomotor al pin 5

pinMode(led_der_d, OUTPUT);
pinMode(led_izq_d, OUTPUT);
pinMode(led_der_a, OUTPUT);
pinMode(led_izq_a, OUTPUT);

while (!Serial) ; // Wait for serial port to be available
if (!rf95.init())
  Serial.println("INICIO FALLIDO");
delay(2000);
}

void loop() {

  D_cm1 = distancia1(20); //lectura de distancia
  D_cm2 = distancia2(20); //lectura de distancia
  D_cm3 = distancia3(20); //lectura de distancia
  D_cm4 = distancia4(20); //lectura de distancia

  String data = String(D_cm1) + "-" + String(D_cm2) + "-" + String(D_cm3) +
  "-" + String(D_cm4);
  int dataLength = data.length(); dataLength++;
  uint8_t total[dataLength];
  //Serial.println(data);
  data.toCharArray(total, dataLength);
  rf95.send(total, dataLength);
  rf95.waitPacketSent();

  uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);

  if (rf95.waitForAvailable(140)) {
    //if (rf95.available()) {
    if (rf95.recv(buf, &len)) {
      String dataTotal = (char*)buf;
      explode(dataTotal);
      start = 1;
    }
    else {
      Serial.println("  RECIVIR FALLIDO");
    }
  }
  else {
    Serial.println("  no hay datos de entrada");
    start = 0;
  }
  // delay(400);
  Serial.print("  ON: ");
  Serial.print(start);
  if (start == 1) {
    angulo = map(X, -45, 55, 1700, 850);
    vel = map(V, 195, 435, 240, 0);
    //delay()

    //Serial.print("  angulo: ");
    //Serial.print(angulo);
    Serial.print("  vel: ");
    Serial.println(vel);
  }
}

```

```

// PITO
if (P <= 300 && Y >= -8 && Y <= 8) {
  analogWrite(Buzzer, salida);
} else {
  analogWrite(Buzzer, 0);
}

//LUCES
if (F1 <= 300 && Y >= -8 && Y <= 8) {
  digitalWrite(led_der_d, HIGH);
} else {
  digitalWrite(led_der_d, LOW);
}
if (F2 <= 300 && Y >= -8 && Y <= 8) {
  digitalWrite(led_izq_d, HIGH);
} else {
  digitalWrite(led_izq_d, LOW);
}
if (F3 <= 350 && Y >= -8 && Y <= 8) {
  digitalWrite(led_der_a, HIGH);
} else {
  digitalWrite(led_der_a, LOW);
}
if (F4 <= 350 && Y >= -8 && Y <= 8) {
  digitalWrite(led_izq_a, HIGH);
} else {
  digitalWrite(led_izq_a, LOW);
}
if (F5 <= 270 && Y >= -8 && Y <= 8) {
  digitalWrite(led_der_d, HIGH);
  delay(25);
  digitalWrite(led_der_d, LOW);
  delay(25);
  digitalWrite(led_izq_d, HIGH);
  delay(25);
  digitalWrite(led_izq_d, LOW);
  delay(25);
  digitalWrite(led_der_a, HIGH);
  delay(25);
  digitalWrite(led_der_a, LOW);
  delay(25);
  digitalWrite(led_izq_a, HIGH);
  delay(25);
  digitalWrite(led_izq_a, LOW);
}

// CONTROL DE LA DIRECCION SERVOMOTOR
if (angulo >= 1150 && angulo <= 1375) {
  ServoMotor.write(1275);
} else {
  ServoMotor.write(angulo);
}

if (angulo >= 1375) { //derecha
  digitalWrite(led_der_d,HIGH);
}
if (angulo <= 1150) { // izquierda
  digitalWrite(led_izq_d,HIGH);
}

```

```

    }

    if (Y >= -5 && Y <= 5) {
        mover(1, 0, 1); // PARO // motor derecha
        mover(2, 0, 1); // PARO // motor izquierda
    }
    if (Y <= -5) {
        mover(1, vel, 0); // ATRAS // motor derecha
        mover(2, vel, 0); // ATRAS // motor izquierda
    }
    if (Y >= 5) {
        mover(1, vel, 1); // DELANTE // motor derecha
        mover(2, vel, 1); // DELANTE // motor izquierda
    }
} else {
    mover(1, 0, 1); // PARO // motor derecha
    mover(2, 0, 1); // PARO // motor izquierda
    ServoMotor.write(1275);
    analogWrite(Buzzer, 0);
}

}

// INICIAMOS SUBPROCESOS FUERA DEL VOID LOOP

//VARIABLES RECIVIDAS
void explode(String req) {
    char str[90];
    req.toCharArray(str, 40);
    char * pch;
    pch = strtok (str, ",");
    while (pch != NULL) {
        turn++;
        if (turn == 1) {
            Serial.print("X: ");
            X = atof(pch);
            Serial.print(X);
        }
        if (turn == 2) {
            Serial.print(" Y: ");
            Y = atof(pch);
            Serial.print(Y);
        }
        if (turn == 3) {
            Serial.print(" VEL FL: ");
            V = atof(pch);
            Serial.print(V);
        }
        if (turn == 4) {
            Serial.print(" PI FL: ");
            P = atof(pch);
            Serial.print(P);
        }
        if (turn == 5) {
            Serial.print(" FL1: ");
            F1 = atof(pch);
            Serial.print(F1);
        }
        if (turn == 6) {
            Serial.print(" FL2: ");
            F2 = atof(pch);
        }
    }
}

```

```

        Serial.print(F2);
    }
    if (turn == 7) {
        Serial.print("  FL3: ");
        F3 = atof(pch);
        Serial.print(F3);
    }
    if (turn == 8) {
        Serial.print("  FL4: ");
        F4 = atof(pch);
        Serial.print(F4);
    }
    if (turn == 9) {
        Serial.print("  FL5: ");
        F5 = atof(pch);
        Serial.print(F5);
    }
    pch = strtok (NULL, ",");
    //delay(100);
}
turn = 0;
}

// MEDICION DE DISTANCIAS
float distancial(int n1) {
    long suma1 = 0;
    for (int i = 0; i < n1; i++) {
        suma1 = suma1 + analogRead(sensor1);
    }
    float adc1 = suma1 / n1;
    float distancia_cm1 = 17569.7 * pow(adc1, -1.2062);
    return (distancia_cm1);
}

float distancia2(int n2) {
    long suma2 = 0;
    for (int i = 0; i < n2; i++) {
        suma2 = suma2 + analogRead(sensor2);
    }
    float adc2 = suma2 / n2;
    float distancia_cm2 = 17569.7 * pow(adc2, -1.2062);
    return (distancia_cm2);
}

float distancia3(int n3) {
    long suma3 = 0;
    for (int i = 0; i < n3; i++) {
        suma3 = suma3 + analogRead(sensor3);
    }
    float adc3 = suma3 / n3;
    float distancia_cm3 = 17569.7 * pow(adc3, -1.2062);
    return (distancia_cm3);
}

float distancia4(int n4) {
    long suma4 = 0;
    for (int i = 0; i < n4; i++) {
        suma4 = suma4 + analogRead(sensor4);
    }
    float adc4 = suma4 / n4;
}

```

```

float distancia_cm4 = 17569.7 * pow(adc4, -1.2062);
return (distancia_cm4);
}

// CONTROL DE VELOCIDAD DE LOS MOTORES POLOLU
void mover(int motor, int velocidad, int direccion) {
  //Vamos a definir la función mover, que va a acciona un motor, fijar su
  velocidad y el sentido de giro. Definimos:
  //motor: llamaremos 1 al motor A, y 2 al motor B
  //velocidad: desde 0 a 255
  //dirección: 0 para el giro en sentido horario (delante), 1 sentido
  antihorario (atras)
  digitalWrite(STBY, HIGH);           //deshabilitar standby para mover
  boolean inPin1 = LOW;                // creamos la variable booleana (solo
  puede ser HIGH/LOW) inpin1
  boolean inPin2 = HIGH;               // y le asignamos el valor LOW. A inPin2
  le damos el valor HIGH
  if (direccion == 1) {
    inPin1 = HIGH;
    inPin2 = LOW;
  }
  if (motor == 1) {
    digitalWrite(AIN1, inPin1);
    digitalWrite(AIN2, inPin2);
    analogWrite(PWMA, velocidad);
  } else {
    digitalWrite(BIN1, inPin1);
    digitalWrite(BIN2, inPin2);
    analogWrite(PWMB, velocidad);
  }
}
void stop() { //ahora si, accionamos el standby
  digitalWrite(STBY, LOW);
}

```